

Programabilni logički kontroleri PLC –osnove kontaktorskih šema, programiranje PLC sa primerima

Industrijski sistemi i protokoli -
predavanja 3 i 4

03/2024

prof Darko Marčetić

1. **Osnove kontaktorskih šema**

(NO I NC, mirno i radno, tasteri, kontakti i PLC bitovi)

2. **Lestvičasti (ladder) dijagram** za programiranje PLC

3. **Blokovi** ladder dijagrama

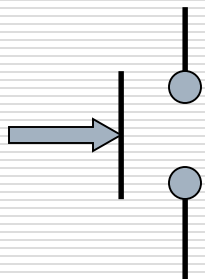
4. **Primeri** Ladder dijagrama

5. **Primena u elektromotornim pogonima**

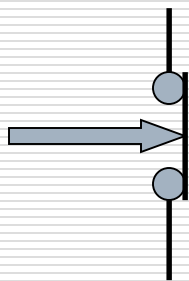
Normalno otvoren (radni) i zatvoren (mirni)?

Tasteri

Šta je radno stanje, akcija, nenormalno stanje kod tastera? **To je pritisak na taster!**

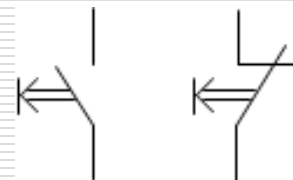


Normalno otvoren (NO-normally open) je otvoren u normalnom neaktivnom stanju (kada nije pritisnut). On je ujedno i **Radni tester** jer **provodi u** radnom stanju, tj. Kada je pritisnut. Njegov grafički simbol je **NO**.



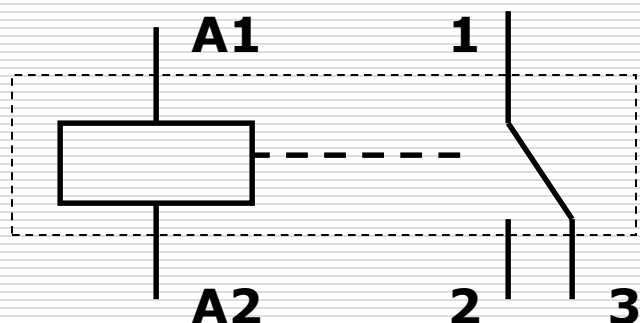
Normalno zatvoren (NC- closed) je zatvoren u normalno neaktivnom stanju (kada nije pritisnut). On je ujedno i **Mirni tester** jer provodi u neaktivnom (mirnom) stanju. Njegov grafički simbol je **NC**.

IEC Oznake za tastere



Normalno otvoren (radni) i zatvoren (mirni)?

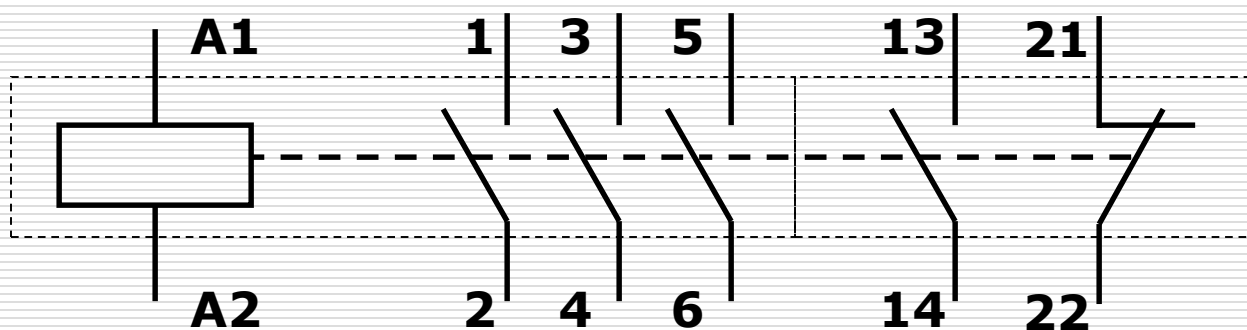
Rele



12 – radni (NO)

13 – mirni (NC)

Kontaktor



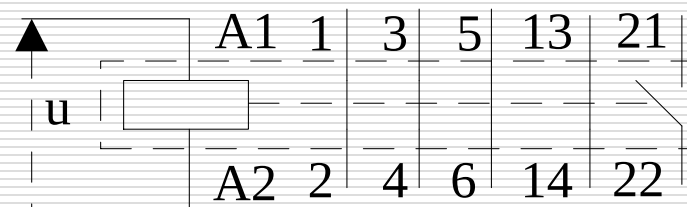
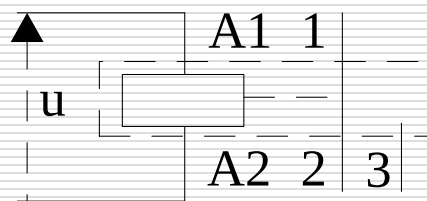
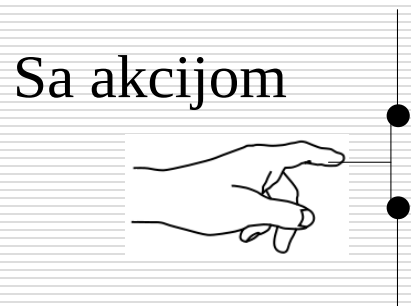
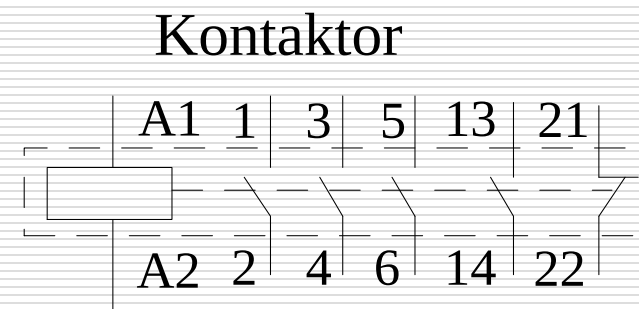
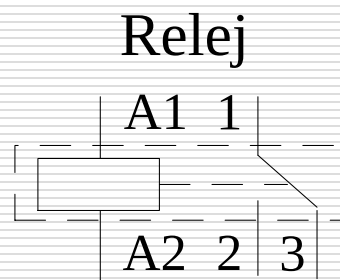
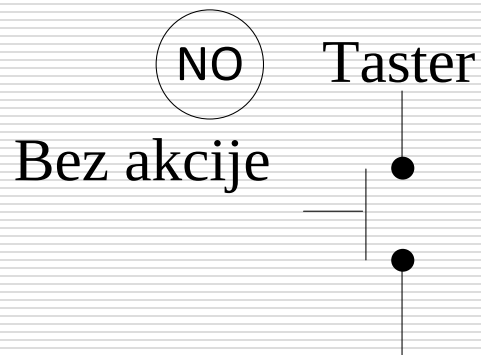
**namotaj
elmagneta
(špulna)**

**glavni
kontakti**

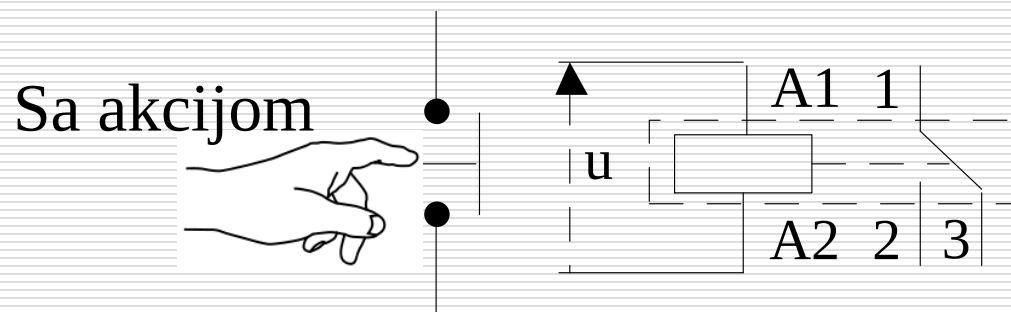
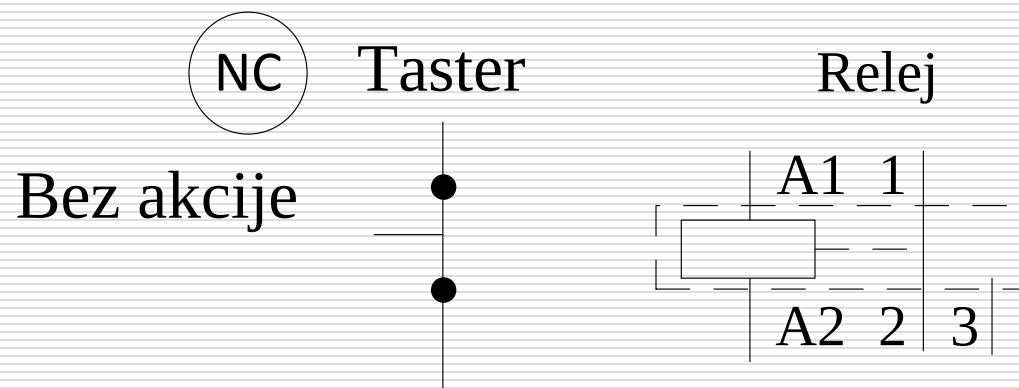
**pomoćni
kontakti**

Obavezno NO !

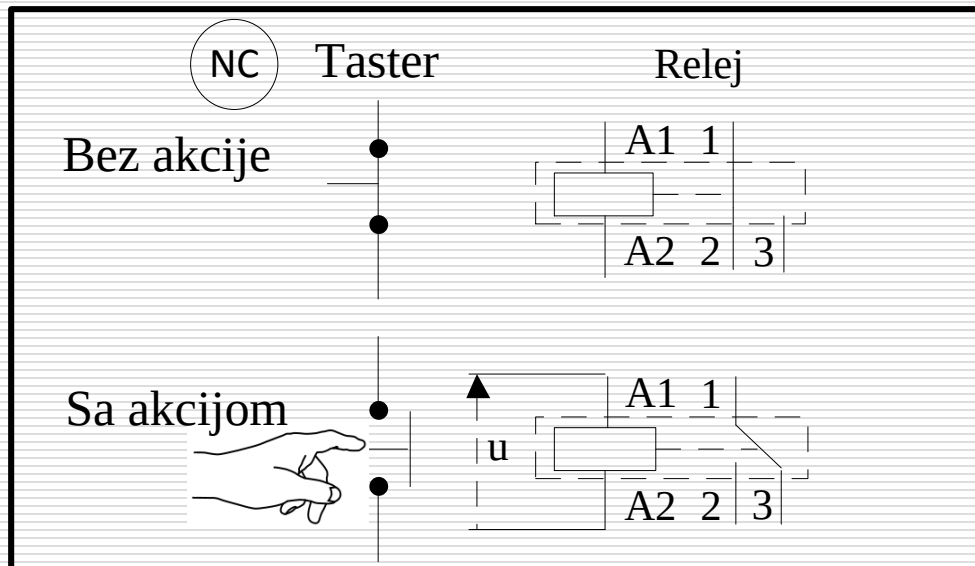
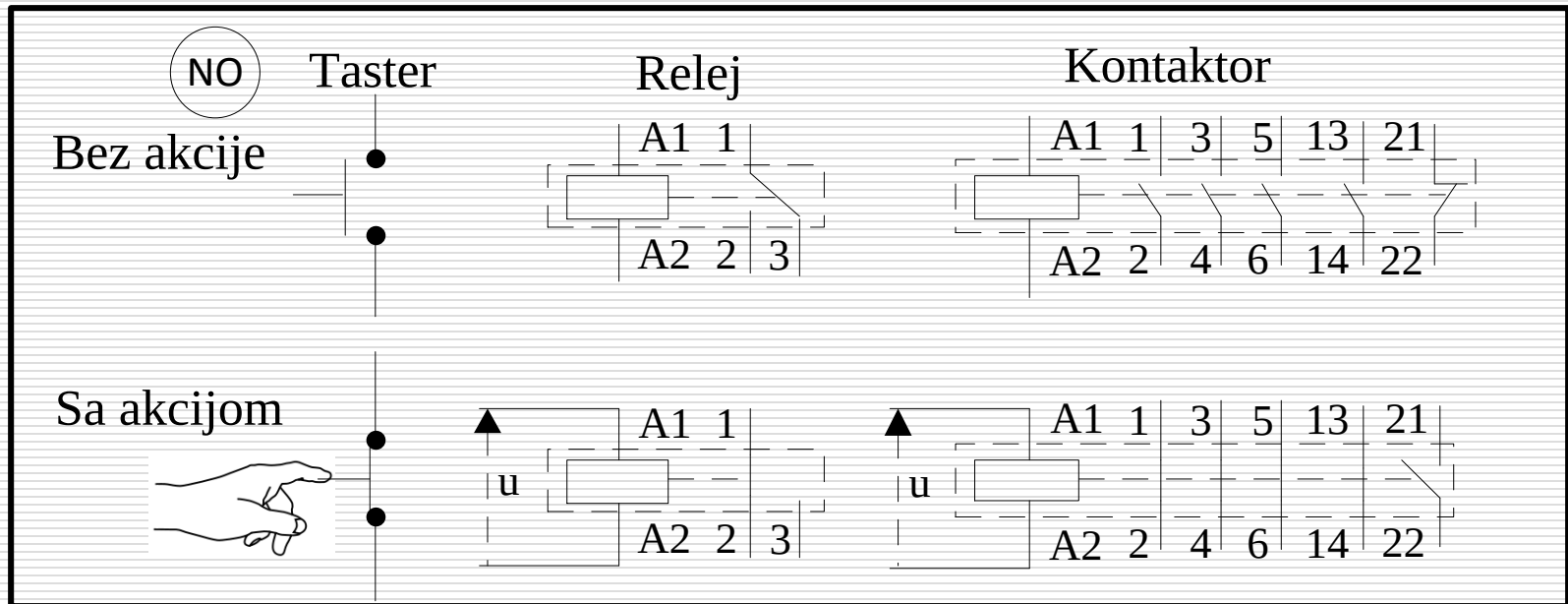
Normalno otvoren (radni)



Normalno zatvoren (mirni)



Normalno zatvoren (NO) i normalno otvoren (NC)



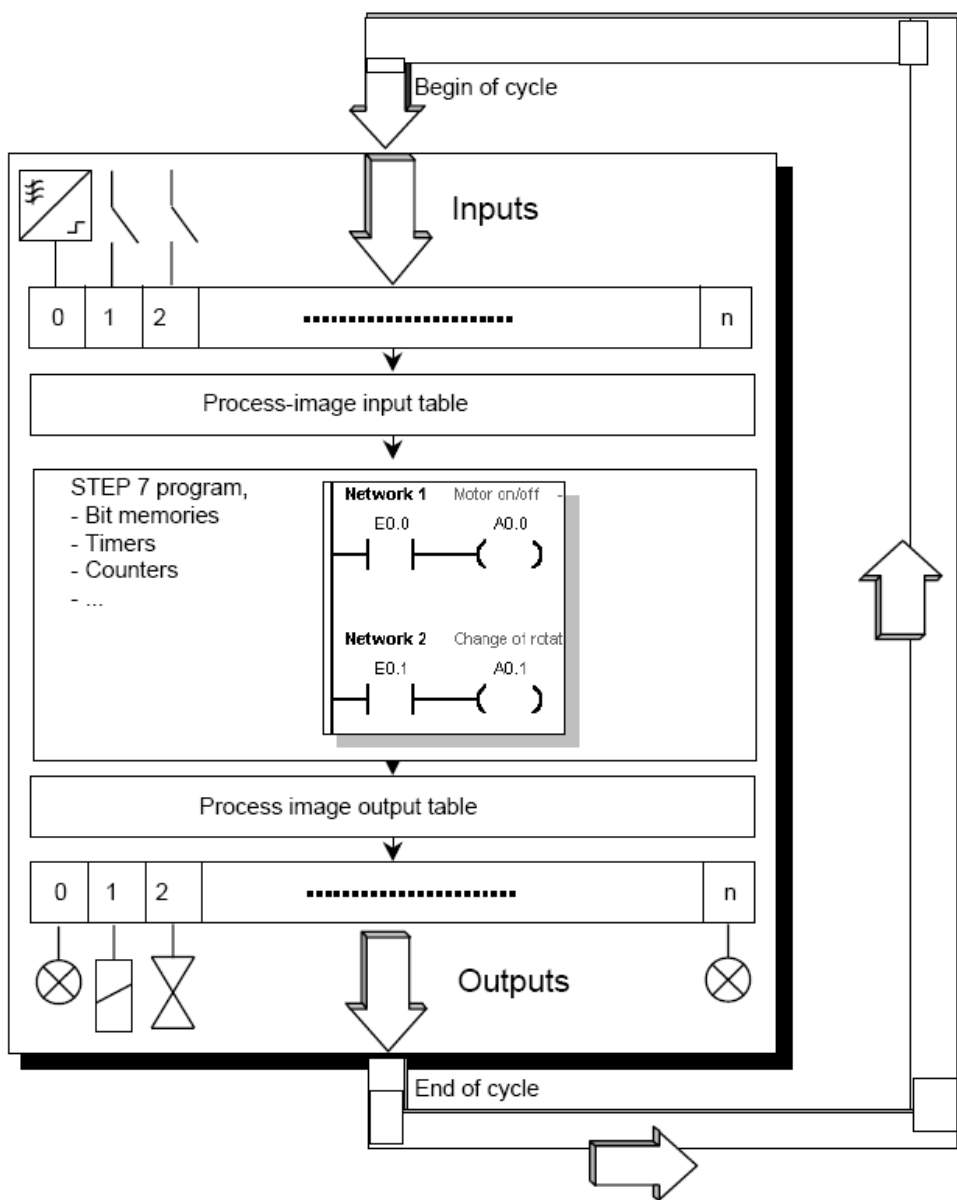
Programiranje putem PLC lestvičastih dijagrama

Umesto relejne šeme PLC koji ciklično izvršava program

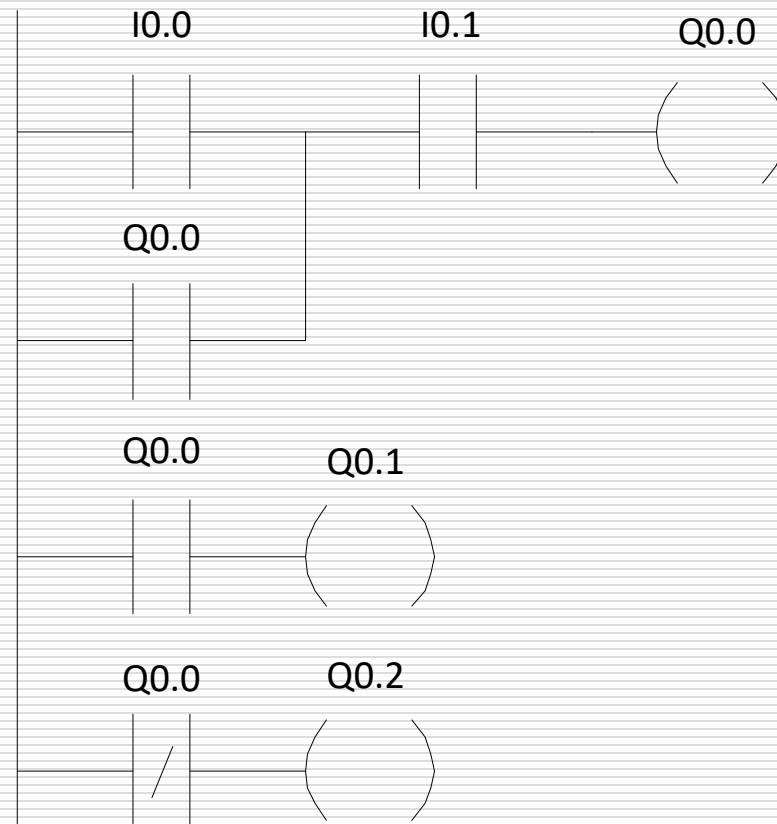
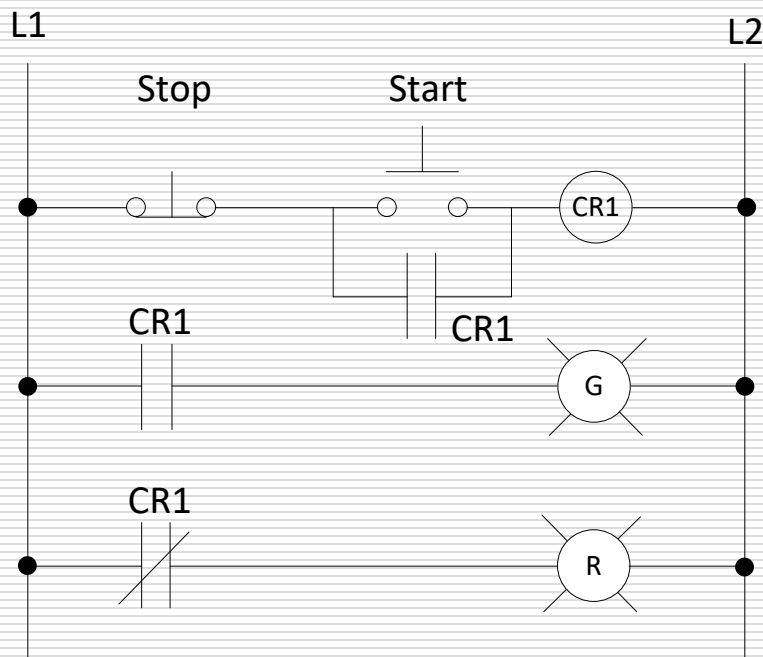
Pročitaj ulaze →

Izvrši program →

Izbaci izlaze →



Prelazak sa relejne šeme na PLC programiranje

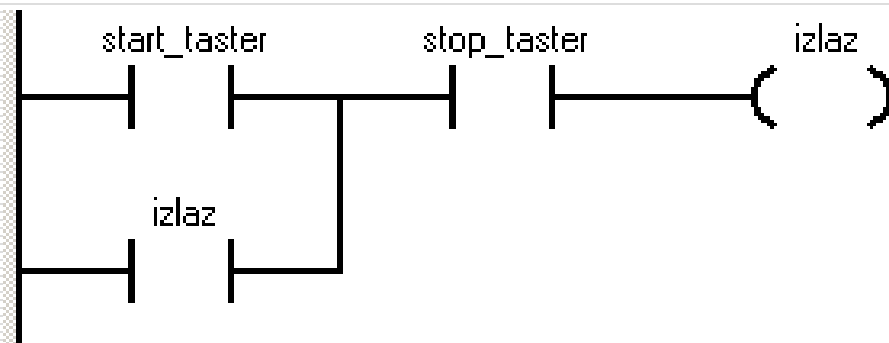


Ista logička funkcija realizovana
a) relejnom šemom

b) lestvičastim dijagramom

Lestvičasta logika i način programiranja

Lestvičasta logika (ladder logic) je način crtanja el. logičkih šema. To je grafički jezik, veoma popular kod PLC. Originalno je izmišljen da zameni relejnu logiku. Ime je dobio jer program podseća na merdevine.



Alternativa

STL (statement list) - Instruction List language

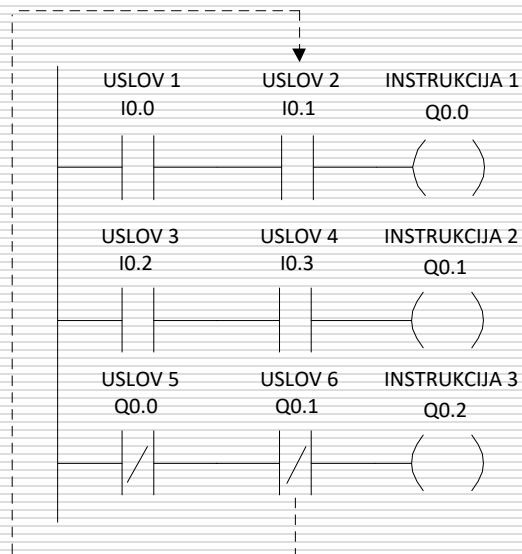
LD	start_taster
O	izlaz
A	stop_taster
=	izlaz

Uslovi i instrukcije u PLC lestvičastom programu



**Ako su zadovoljeni USLOV1 i USLOV2
INSTRUKCIJA 1 će se izvršiti**

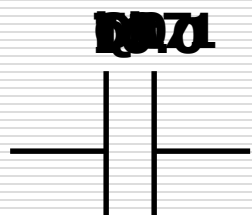
**U datom primeru ako su I0.0 i I0.1 = 1
izlaz Q0.0 će biti 1 , tj. 24V.**



**Instrukcije se izvršavaju redom. Nakon
tačno određenog vremena , scan time,
ciklus se ponavlja.**

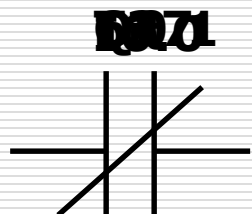
Normalno otvoren i zatvoren uslov

Uslovi u PLC programu koje kontrolišu bitovi



Šta je radno stanje jednog bita ? **To je stanje 1, stanje ON !**

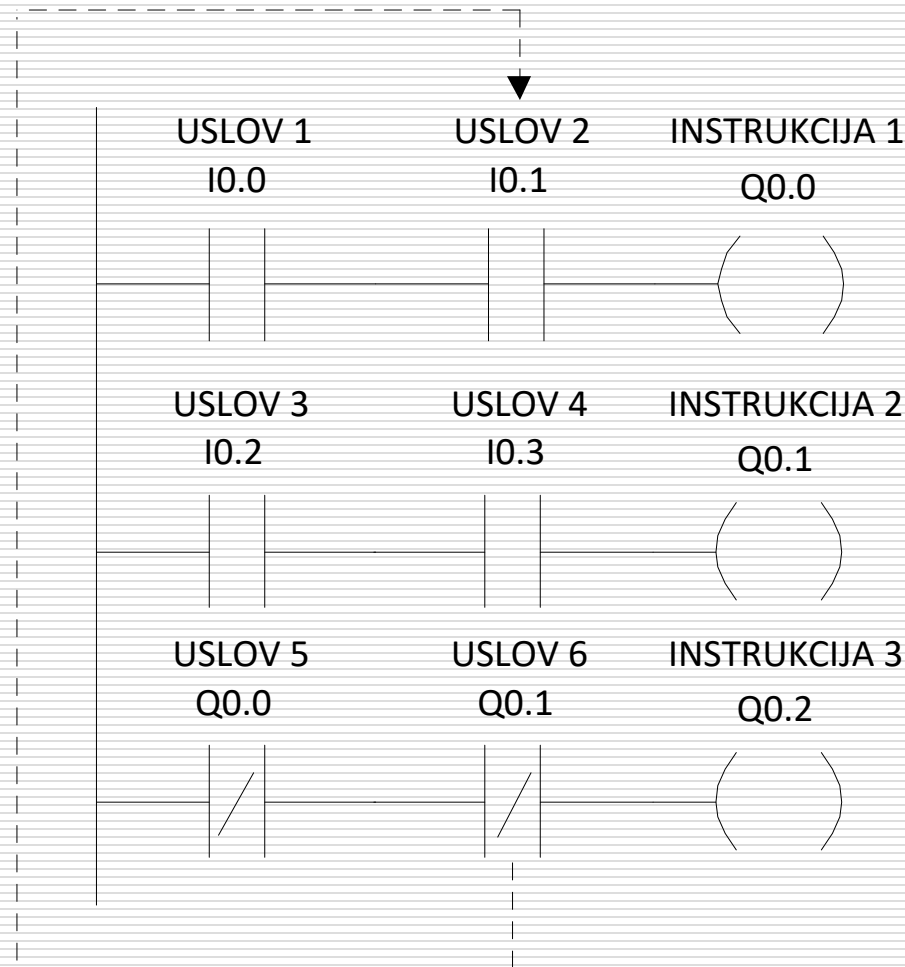
NO uslov vodi kada je bit koji ga kontroliše **1 (JEDAN)**
→ **normalno otvoren uslov.**



NC uslov vodi kada je bit koji ga kontroliše **0 (NULA)**
→ **normalno zatvoren uslov**

Bit može biti ulaz (na primer I0.0) koji je 1 ako je napon na ulazu, ili izlaz (na primer Q0.0) koji je 1 ako je napon na izlazu, ili čak i samo bit iz memorije (na primer M0.0) koji je 1 ako je prethodno setovan ili 0 ako je prethodno resetovan.

Šta sve može da se koristi u lestvičastom programu?



PLC: Osnovni blokovi lestvičastog dijagrama

PLC S7 osnovni Ladder blokovi - uslovi

USLOVI

- ❑ Mogu biti NO i NC
- ❑ mogu biti kontrolisani raznim bitovima: ulaznim I0.0, izlaznim bitovima Q0.0, memorijskim bitovima M0.0, tajmerskim T37, brojačkim C30...

INSTRUKCIJE

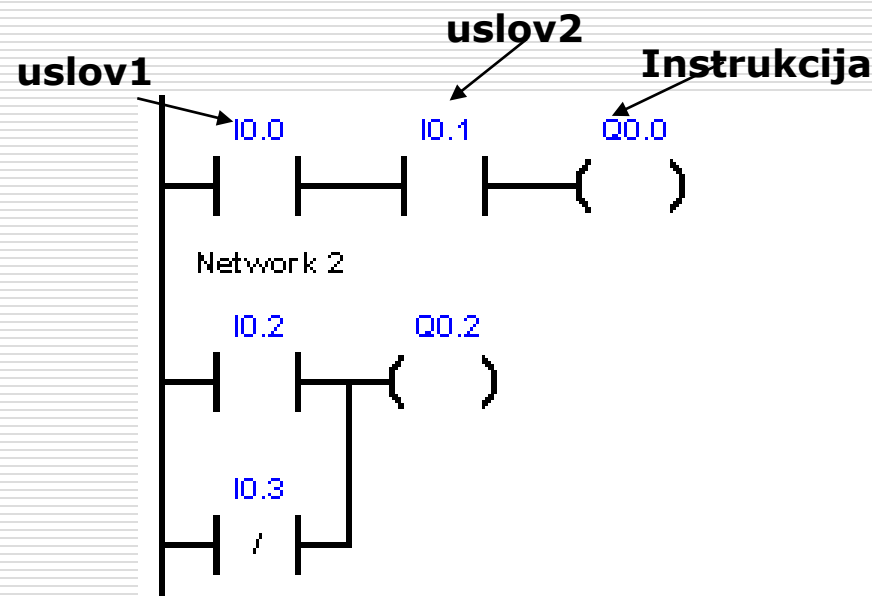
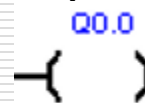
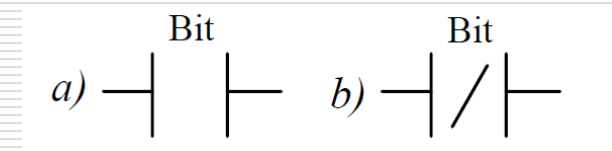
- ❑ Izvršavaju se ako su uslovi za njihovo izvršenje zadovoljeni
- ❑ Instrukcija može biti prosto uključenje izlaza, ali i neka složena operacija

Primer korišćenja uslova i instrukcija:

Na osnovu trenutnog stanja bitova koji ih kontrolišu izabrana kombinacija uslova definiše da li će instrukcija biti omogućena ili ne.

$Q0.0 = I0.0 \text{ i } \overline{I0.1}$

$Q0.2 = I0.2 \text{ ili } I0.3$



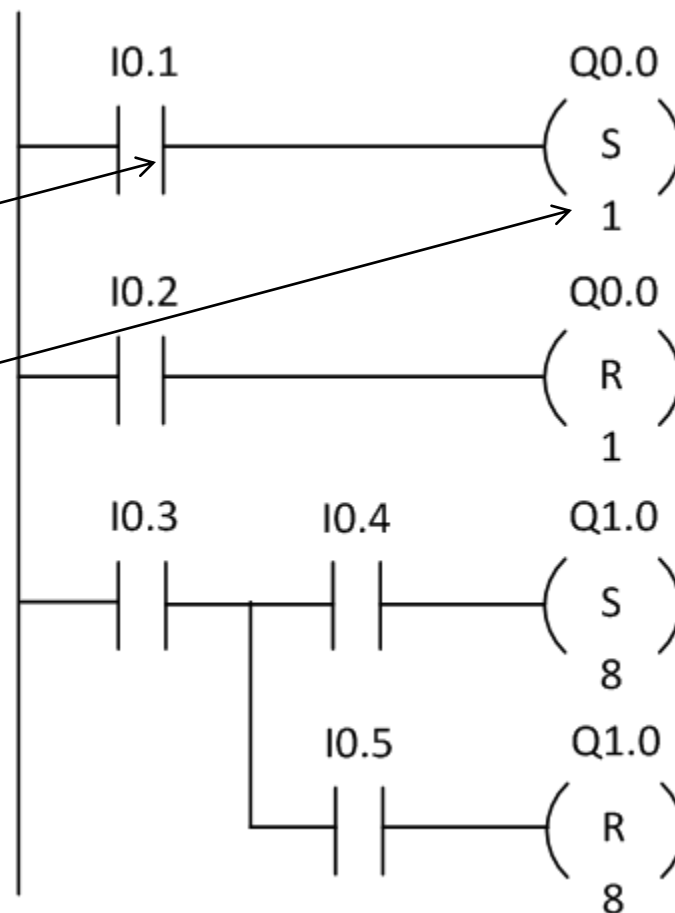
PLC S7 osnovni Ladder blokovi– set/reset

SET (S) i RESET (R)

instrukcija vrše setovanje (uključenje; postavljanje na nivo 1), odnosno resetovanje (isključenje; spuštanje na nivo 0) određenog broja pozicija (do 255), počevši od lokacije određenog adresnog bita.

Ipak, tipično se setuje ili resetuje jedna pozicija, tj. jedan bit.

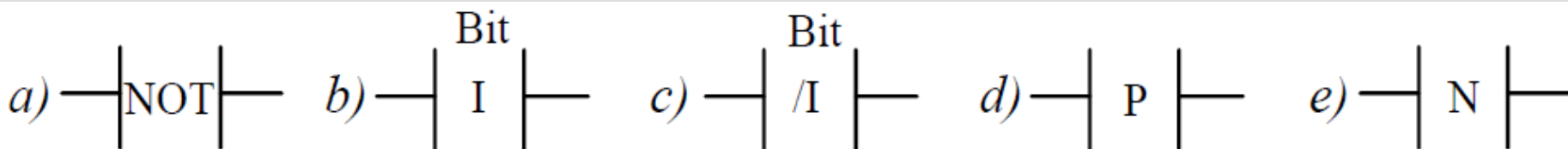
Aktivacija ulaza I0.1 u prvoj liniji instrukcija omogućava izvršenje SET instrukcije koja setuje jednu bit lokaciju, tj. izlaz Q0.0. Aktivacija ulaza I0.2 u drugoj liniji instrukcija omogućava izvršenje RESET instrukcije koja isti taj izlaz resetuje. Ukoliko ulazi I0.1 i I0.2 nisu više aktivni izlaz Q0.0 pamti svoje prethodno stanje i ostaje u stanju set ili reset.



PLC S7 osnovni Ladder blokovi– spec uslovi

Specifični uslovi

- a) NOT uslov menja stanje svog izlaza (toggle).
- b) Direktna NO instrukcija, menja izlaza odmah, ne čekajući kraj sken ciklusa.
- c) Direktna NC instrukcija, menja izlaza odmah, ne čekajući kraj sken ciklusa.
- d) Instrukcija prelaska, reaguju na pozitivnu ivicu signala na ulazu (0 na 1),
- e) Instrukcija prelaska, reaguju na negativnu ivicu signala na ulazu (1 na 0),

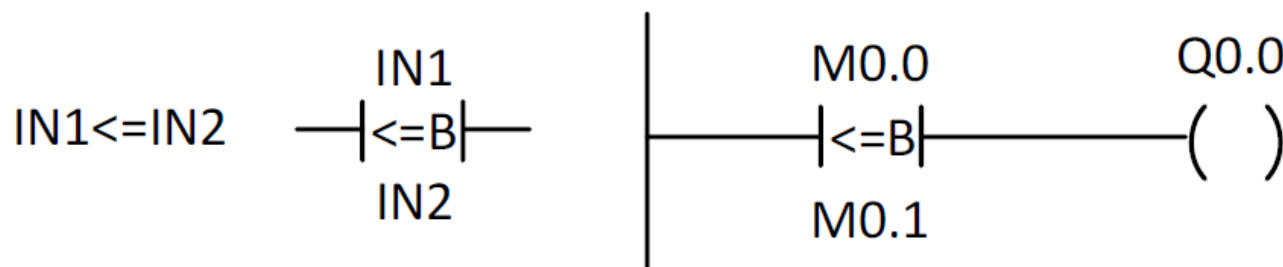


PLC S7 osnovni Ladder blokovi– poređenje

Instrukcije poređenja

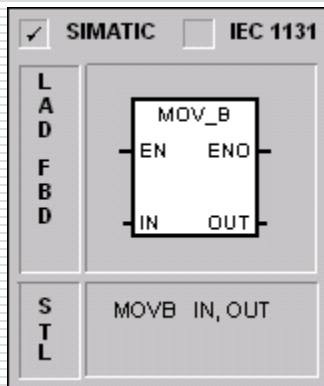
Instrukcije poređenja se koriste za poređenje dve vrednosti. Ukoliko je rezultat poređenja istinit izvršava se naredba koja sledi iza instrukcije poređenja. Neke od mogućih relacija poređenja su: ==, >, <, >=, <= i < >. Ukoliko se porede bajtovi (*Compare Byte – B*) tada je rezultat te operacije uvek neoznačen (npr. ==B). Rezultat operacije poređenja je označen ako se porede celobrojne vrednosti (*Compare Integer – I*), vrednosti proširene tačnosti (*Compare Double Word – D*) i realni brojevi (*Compare Real – R*).

Na slici 5.32 desno je dat i primer. Ukoliko je vrednost u lokaciji M0.0 manja ili jednaka vrednosti u M0.1 izlaz Q0.0 će biti aktivan, a ako je veća, izlaz Q0.0 je neaktivan.



PLC S7 osnovni Ladder blokovi– MOV

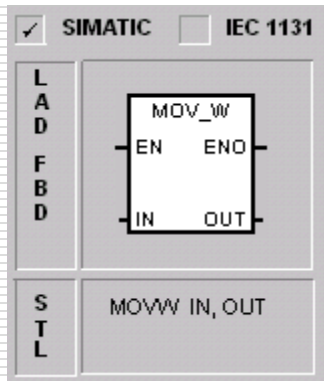
DODELA VREDNOSTI MOV instrukcija



Move Byte (MOVB) instrukcija kopira broj u ulaznom bajtu (IN) u izlazni bajt (OUT) bez promene vrednosti.



Ovaj primer upisuje broj 102 u lokaciju VB10



**Move Word (MOVW) instrukcija kopira broj u ulaznom reči (IN) u izlaznu reč (OUT) bez promene vrednosti.
Word = 16 bitova = 2 bajta**

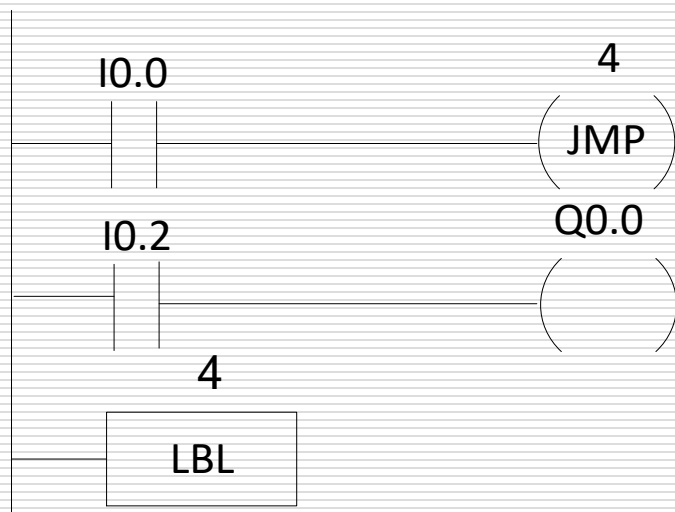
Ove dve instrukcije se često za podešavanje parametara dodatnih modula. Na primer, podešavanje nekog komunikacionog modula zahteva 9600 brzinu prenosa u lokaciji X koja definiše baudrate. Tada, u registar X preko MOVEW postavimo 9600 tako što IN povežemo sa 9600, a na OUT adresu registra X.

PLC S7 osnovni Ladder blokovi– JMP and LBL

SKOK

Kontrola toka programa , JMP and LBL

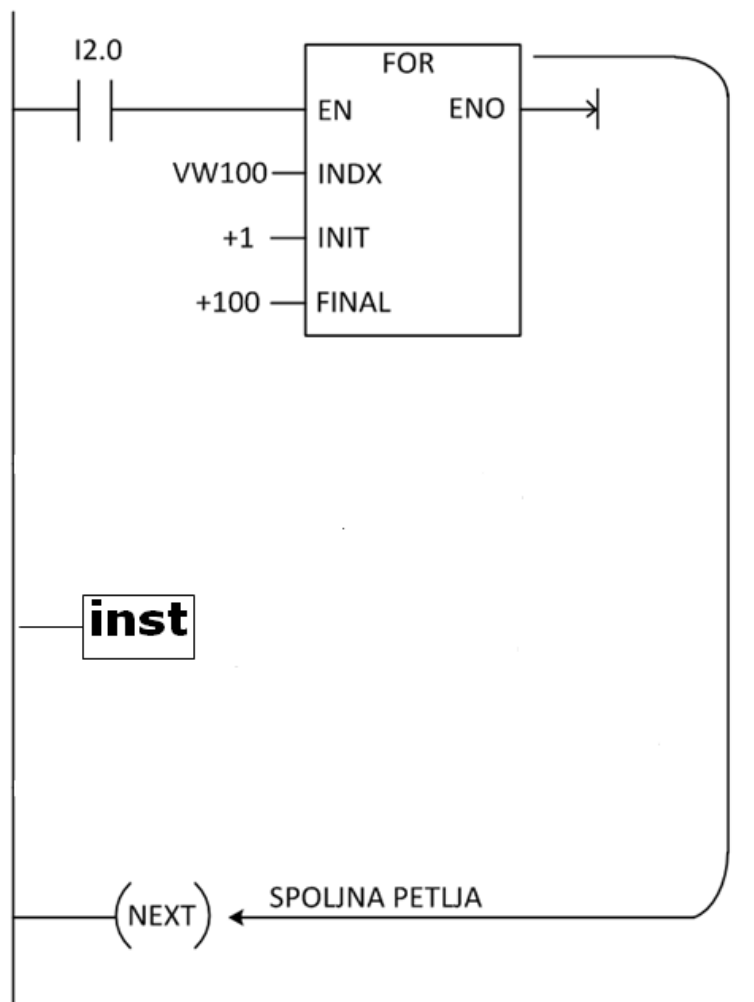
Jump to label (JMP) instrukcija predstavlja grananje (skok) ka određenoj liniji tj. labeli (LBL) unutar programa. Ukoliko je aktivan NO uslov kontrolisan bitom I0.0, dakle ako je na ulazu I0.0 napon od 24V, omogućava se izvršenje *Jump to Label* instrukcije. U datom primeru omogućena JMP instrukcija izvršava grananje programa ka 4-toj labeli (poziciji) u programu. Isti broj 4 se nalazi iznad instrukcije JMP i njoj odgovarajuće labele. Dakle, ako je I0.0=1, u datom PLC ciklusu se preskače druga instrukcijska linija programa i ona se neće izvršiti.



PLC S7 osnovni Ladder blokovi– FOR and NEXT

PETLJA

Kontrola toka programa , FOR and NEXT

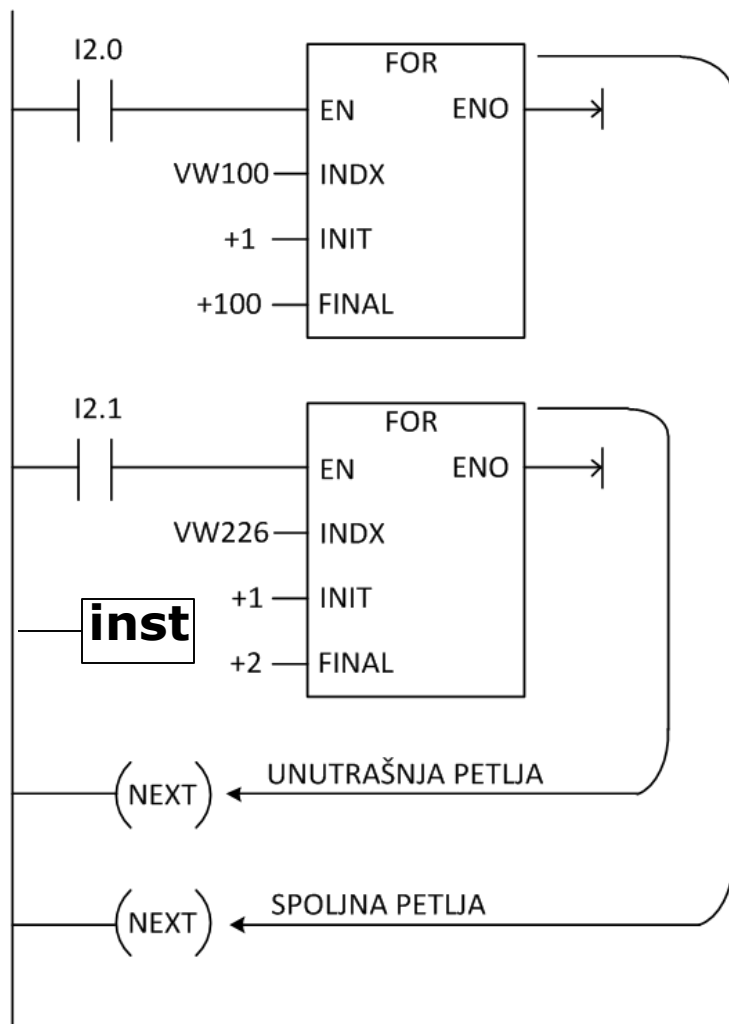


FOR naredba izvršava instrukcije koje se nalaze između reči FOR i NEXT. Za ovu instrukciju je potrebno zadati indeksnu vrednost (u primeru mem lokacija VW100), početnu (1) i krajnju vrednost (100). Primer će 100 puta izvršiti spoljnu petlju a u svakoj spoljnoj i dva puta unutrašnju, dakle ukupno 200 prelazaka i instrukcija inst bi se izvršila 200 puta u svakom PLC ciklusu.

PLC S7 osnovni Ladder blokovi– FOR and NEXT

PETLJA

Kontrola toka programa , FOR and NEXT



FOR naredba izvršava instrukcije koje se nalaze između reči FOR i NEXT. Za ovu instrukciju je potrebno zadati indeksnu vrednost (u primeru mem lokacija VW100), početnu (1) i krajnju vrednost (100). Primer će 100 puta izvršiti spoljnu petlju a u svakoj spoljnoj i dva puta unutrašnju, dakle ukupno 200 prelazaka i instrukcija inst bi se izvršila 200 puta u svakom PLC ciklusu.

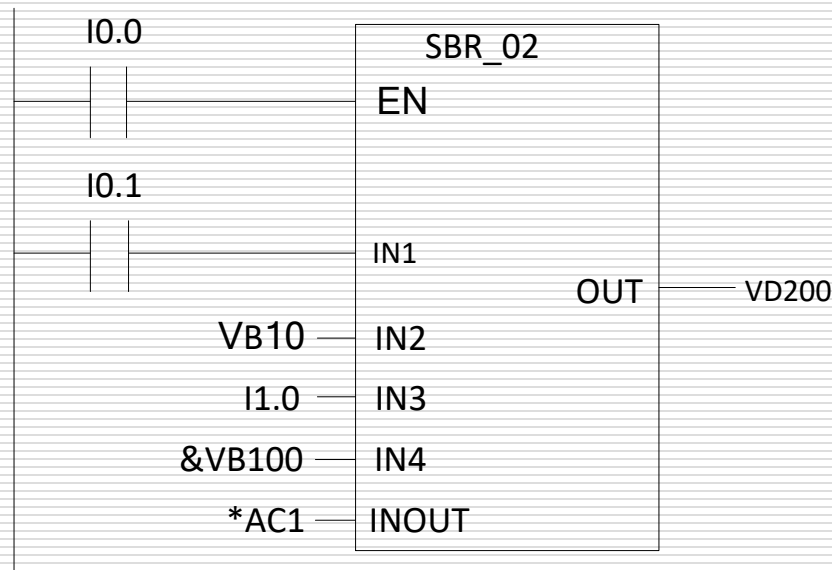
PLC S7-200 – potprogram (subroutine)

POTPROGRAM

Kontrola toka programa

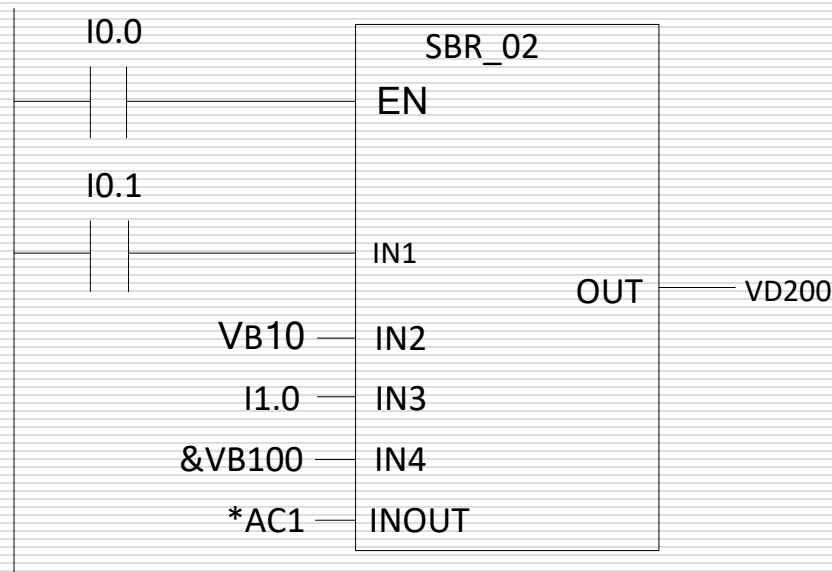
Glavni PLC program može da inicira izvršenje potprograma. Potprogram se izvršava do kraja i nakon toga predaje kontrolu glavnom programu nad mrežom sa koje je potprogram bio pozvan.

Potprogram se često koristi pri radu sa komunikacionim ili nekim drugim modulima. PLC programer samo preuzme potprogramski blok koji inicijalizuje ili kontroliše neki od serijskih portova i na taj način, bez potrebe za razumevanjem svih detalja, kontroliše rad tih portova, šalje ili prima podatke, itd.



PLC S7-200 – potprogram (subroutine)

Tip parametra	Objašnjenje
IN	Parametri potprograma. Ako upravlja adresom (kao što je VB10), vrednost bita na određenoj lokaciji se uvodi u potprogram. Ako je podatak konstanta (16#1234), ili neka adresa (&VB100), vrednost konstante ili adrese je provedena u potprogram.
IN_OUT	Provodi vrednost sa određene lokacije parametra i rezultujuću vrednost iz potprograma vraća na istu lokaciju. Konstante (kao što je 16#1234) i adrese (kao što je &VB100) ne mogu biti ulazno/izlazni parametri.
OUT	Rezultujuću vrednost iz potprograma vraća na lokaciju određenog parametra.
TEMP	Bilo koja lokalna memorija koja nije korišćena za prosleđivanje parametara može biti upotrebljena za privremeno skladištenje unutar potprograma.



PLC S7-200 – Prekid (interrupt)

Informativno

S7-200 PLCs incorporate instructions for use with interrupts.

Interrupts are used to initiate a specific, short PLC program segment, called an interrupt routine, when an internal or external event occurs. After the interrupt routine has been executed, control is returned to the main program.

Three types of interrupts are supported by S7-200 PLCs,

- communication port interrupts,**
- I/O interrupts,**
- timebased interrupts.**

Communication port interrupts are used to control a communication port operated in Freeport mode.

I/O interrupts are used to respond quickly to high-speed I/O transitions, such as those associated with high-speed counters or pulse train outputs.

Time-based interrupts allow the user program to execute an interrupt routine on a cyclic basis.

Each of these types of interrupts has an associated priority that determines which interrupt is processed first in the event that two or more interrupts are requested at the same time. Communication port interrupts have the highest and time-based have the lowest priority.

1. Prekid (interrupt, u PLC svetu često zvan događaj ili event) se opslužuje nezavisno od periodičnog PLC ciklusa skeniranja.

2. Kod prekida treba definisati dve stvari

- **Izvor prekida**
- **Podprogram koji treba izvršiti kada se desi prekid**

3. Te dve stvari treba spojiti, ATTACH blok !

ATTACH blok kaže pridruži ovaj podprogram (interrupt routine) sa onim interruptom ili događajem (event)

4. Ponekad treba redefinisati akciju , onda prvo moramo razdvojiti rutine od događaja, za to služi DTCH blok

Naravno, događaj (prekid) treba dozvoliti (ENI) ili po potrebi zabraniti (DISI). U startu je većina zabranjena.

PLC S7-200 – Prekid (interrupt)

Informativno

ATCH traži i ime podprograma za izvršenje, koji treba povezati sa tim događajem

The Enable Interrupt (ENI) instruction globally enables processing of all attached interrupt events.

The Disable Interrupt (DISI) instruction disables processing of all interrupt events.

ATCH traži broj događaja tj prekida

The Attach Interrupt (ATCH) instruction associates an interrupt event (EVNT) with an interrupt routine number (INT), and enables the interrupt event.

The Detach Interrupt (DTCH) instruction disassociates an interrupt event (EVNT) from all interrupt routines, and disables the interrupt event.

DTCH odvaja događaja od svih rutina. Ne mora da zna koja je bila priljučena tom događaju, sve ubija !

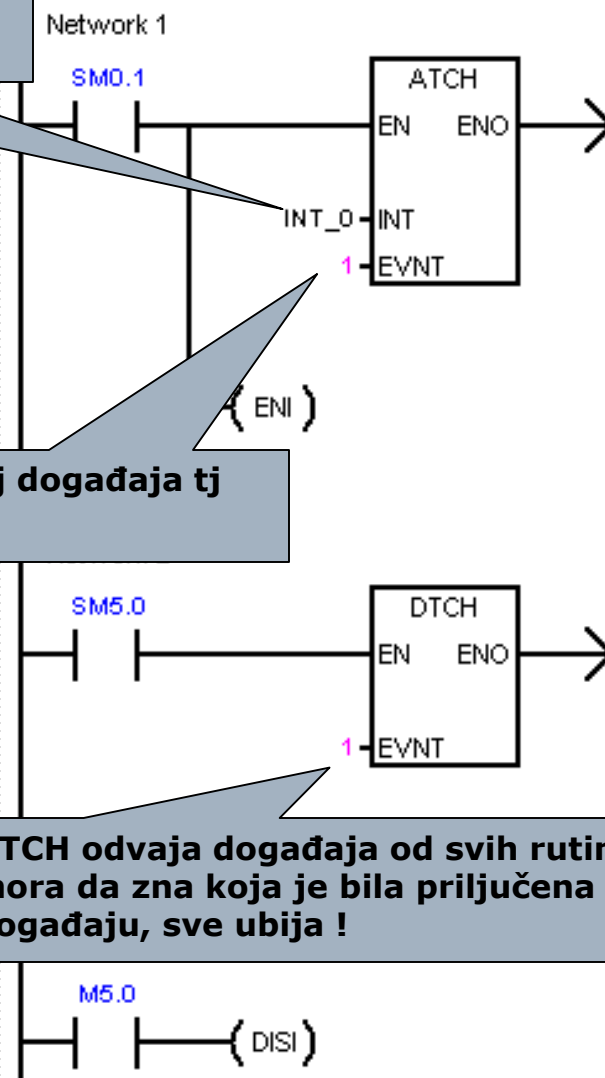


**Tabela Event-a
za PLC S200
(nepotpuna)**

Event	Description	CPU 221 CPU 222	CPU 224	CPU 224XP CPU 226
0	I0.0 Rising edge	Y	Y	Y
1	I0.0 Falling edge	Y	Y	Y
2	I0.1 Rising edge	Y	Y	Y
3	I0.1 Falling edge	Y	Y	Y
4	I0.2 Rising edge	Y	Y	Y
5	I0.2 Falling edge	Y	Y	Y
6	I0.3 Rising edge	Y	Y	Y
7	I0.3 Falling edge	Y	Y	Y
8	Port 0 Receive character	Y	Y	Y
9	Port 0 Transmit complete	Y	Y	Y
10	Timed interrupt 0 SMB34	Y	Y	Y
11	Timed interrupt 1 SMB35	Y	Y	Y
12	HSC0 CV=PV (current value = preset value)	Y	Y	Y
13	HSC1 CV=PV (current value = preset value)		Y	Y
14	HSC1 Direction changed		Y	Y
15	HSC1 External reset		Y	Y
16	HSC2 CV=PV (current value = preset value)		Y	Y
17	HSC2 Direction changed		Y	Y
18	HSC2 External reset		Y	Y
19	PLS0 PTO pulse count complete interrupt	Y	Y	Y
20	PLS1 PTO pulse count complete interrupt	Y	Y	Y
21	Timer T32 CT=PT interrupt	Y	Y	Y

PLC S7-200 – Osnovni matematički blokovi

Floating point matematika

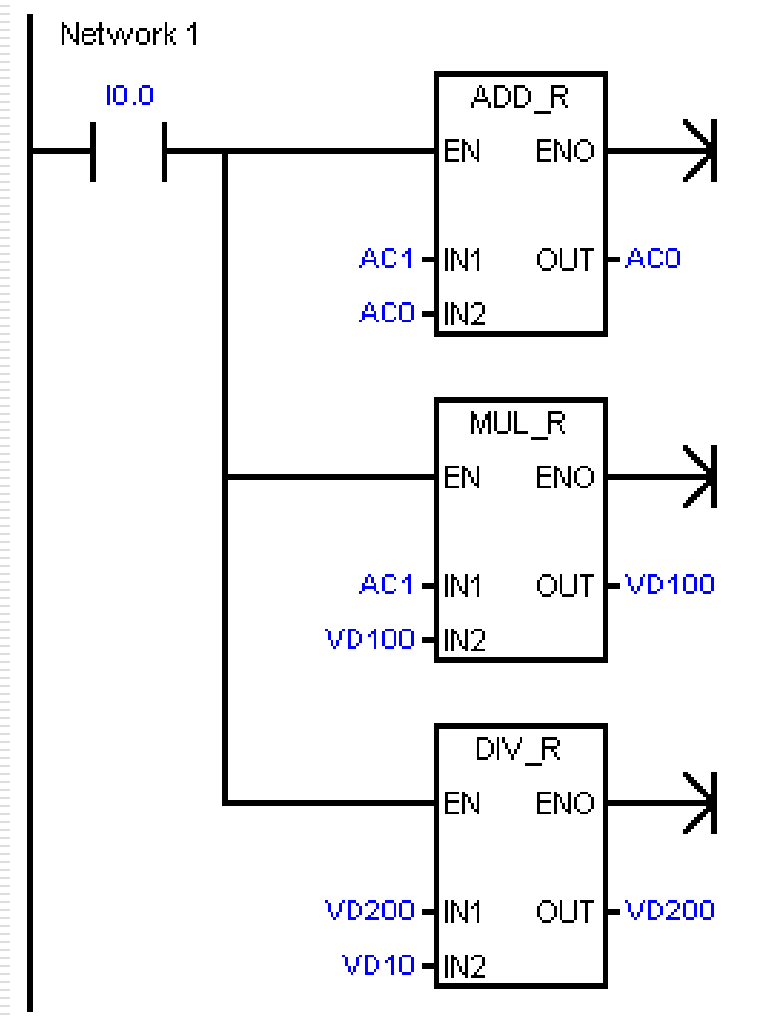
IN1 + IN2 = OUT

IN1 - IN2 = OUT
(nije prikazano)

IN1 * IN2 = OUT

IN1 / IN2 = OUT

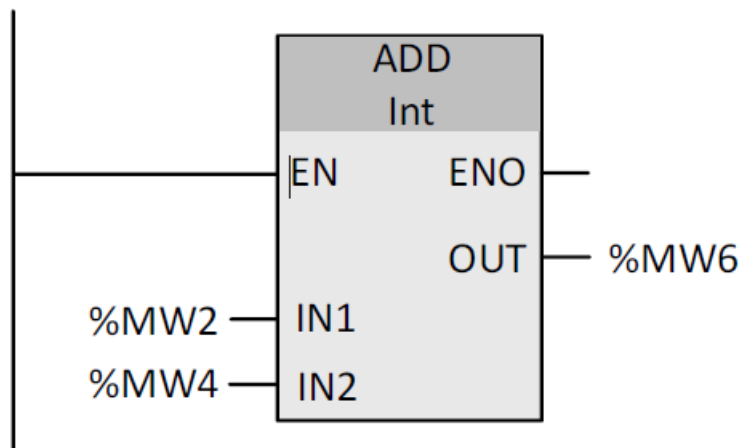
Itđ..



PLC S7-1200 – Osnovni matematički blokovi

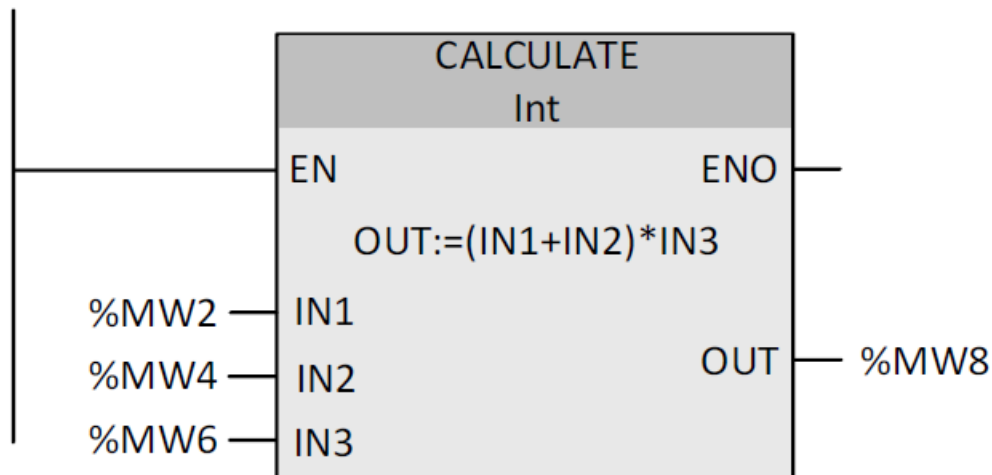
Blokovi ADD, SUB, MUL, DIV, MOD

$$\text{OUT} = \text{IN1} + \text{IN2}$$



Blok CALCULATE

$$\text{OUT} = (\text{IN1} + \text{IN2}) * \text{IN3}$$

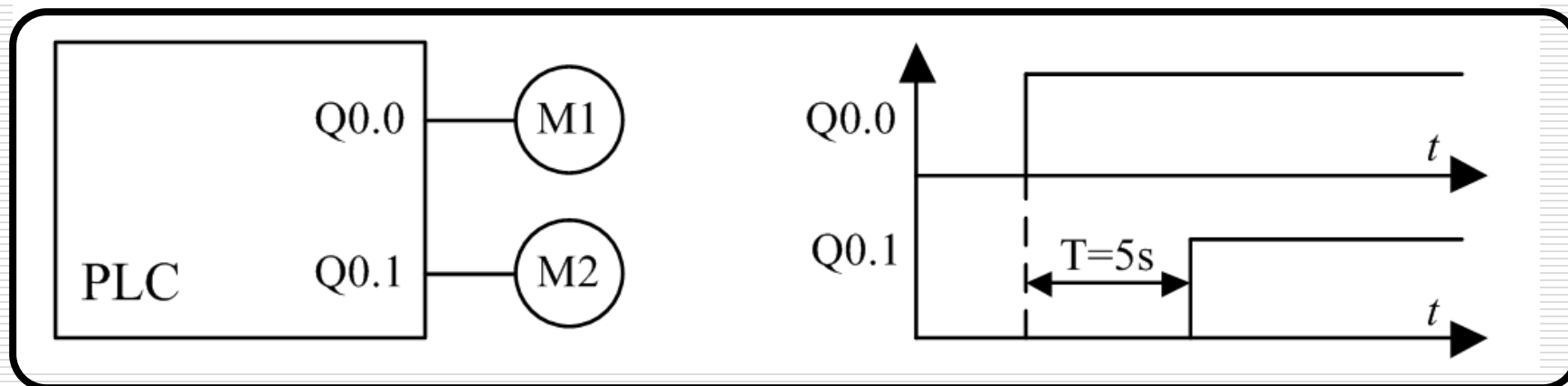


Blokovi za merenje vremena (Timer blokovi)

Čemu služe Tajmer blokovi u PLC programu ?

Postavljanje kontrolnih signala u tačno definisanim vremenskim trenucima je jedan od važnijih zadataka tokom upravljanja industrijskim sistemima.

Primer: motor broj 2 treba da se uključi tačno 5 sekundi kasnije u odnosu na start motora broj 1.



Da bi PLC ovaj zadatak precizno izvršio neophodno je da poseduje blok za merenje vremena. U tu svrhu PLC poseduju brojačke *Timer* blokove koji mere vreme tako što broje impulse sistemskog sata čija je perioda poznata.

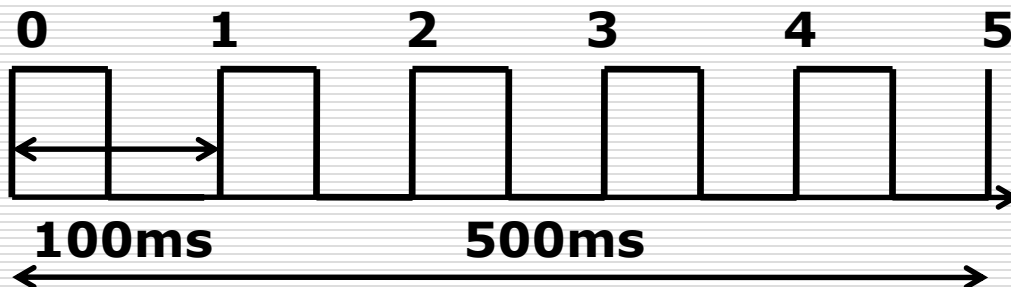
Koje Tajmer blokove PLC poseduje ?

PLC podržava dve osnovne grupe *Timer* blokova:

1. tajmeri koji kasne start (uključenje), tj. sa zakašnjenjem propuštaju rastuću ivicu na svojem izlazu,
2. tajmeri koji kasne isključenje (stop), tj. sa zakašnjenjem propuštaju opadajuću ivicu na svom izlazu.

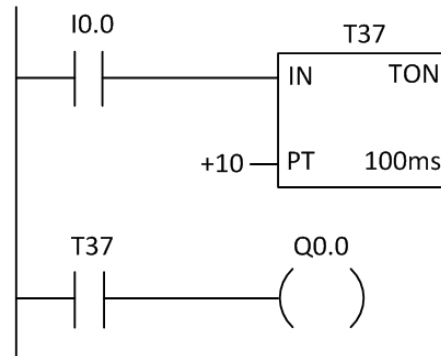
Timer blokovi postoje sa različitim vremenskim bazama, tj, periodama impulsa koje mere: 1ms , 10ms ili 100ms ...

Timer broji impulse čija je učestanost (perioda) unapred poznata. Rezultat toga je da je broj impulsa koji Timer izbroji proporcionalan proteklom vremenu.



Tajmer koji kasni start *On-Delay Timer* (TON)

On-Delay Timer služi da odloži uključenje nekog uređaja. Na primer, nakon pritiska start tastera treba prvo uključiti pumpu za hlađenje, a tek onda motor koji ta pumpa hladi. U tom slučaju pumpu treba startovati odmah, ali start motora treba odložiti za potrebno vreme.

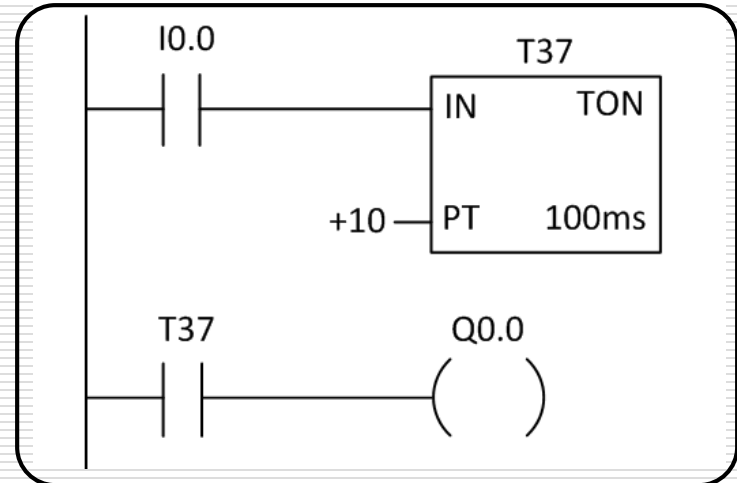


TON blok broji ima sledeće elemente:

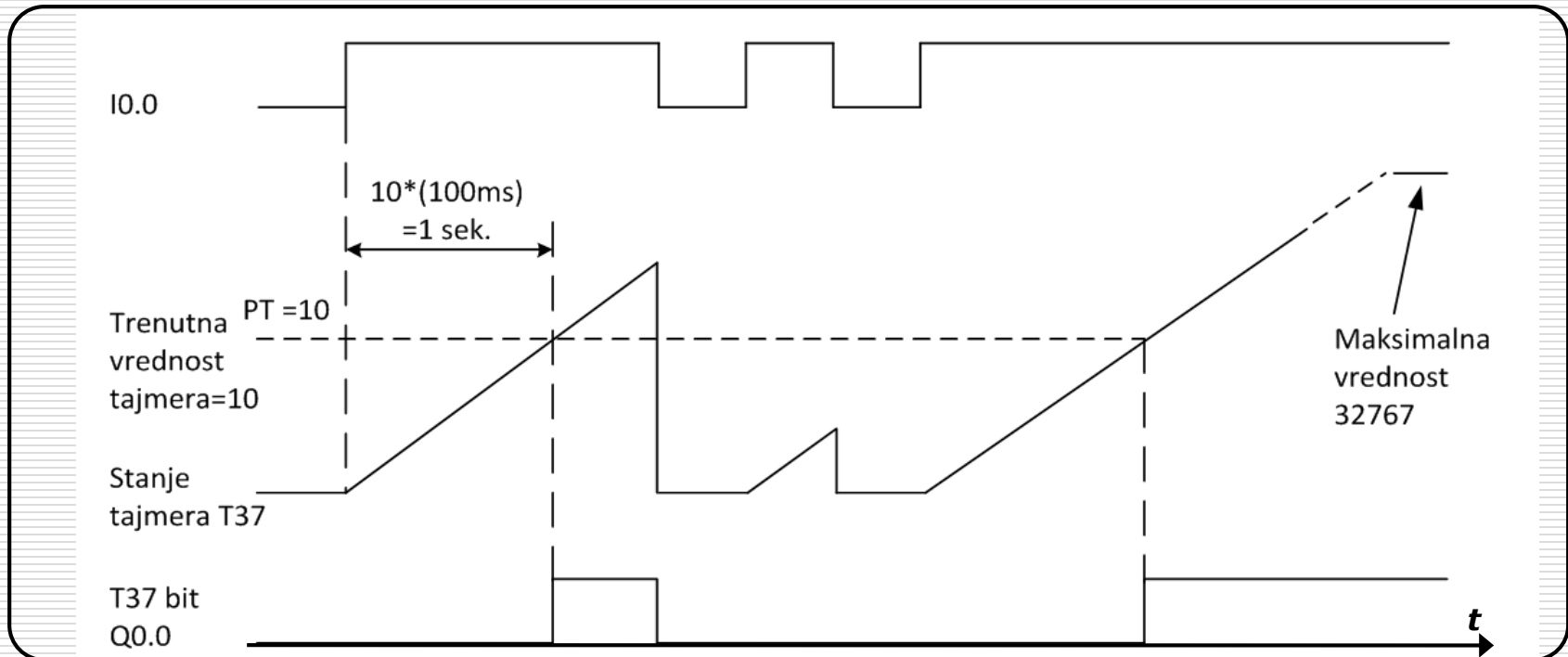
1. Aktivacioni ulaz (IN) koji mu omogućava rad. U datom primeru blok TON meri vreme kada se aktivira uslov ulaza I0.0.
2. Rezoluciju ili vremenski period koji odgovara jednom impulsu. U datom primeru je uzet T37 čija je vremenska rezolucija jednaka 100ms.
3. Unapred postavljeno vreme koje se meri, PT (*Preset Time*). U datom primeru PT je postavljeno na 10. Realizovano vreme pored broja PT zavisi i od rezolucije tajmera, u datom primeru $10 * 100 \text{ ms} = 1 \text{ s}$.
4. Bit koji taj *Timer* blok kontroliše, u datom primeru je uzet *Timer* broj 37 koji kontroliše bit T37.

Tajmer koji kasni start *On-Delay Timer* (TON)

Ako je IN aktiviran, TON počinje da broji od nule. Dok je trenutna izbrojena vrednost manja od postavljenog vremena PT, T37 bit je nula. Ako je izbrojena vrednost jednaka ili veća od PT bit T37 je uključen (T37=1). U datom primeru PLC programa aktiviranje T37 bita omogućuje izvršenje instrukcije Q0.0, tj. aktivira izlaz. Dakle, pojava pozitivne ivice na Q0.0 je zakašnjena za 1 sekund ($10 \cdot 100\text{ms}$) u odnosu na I0.0.

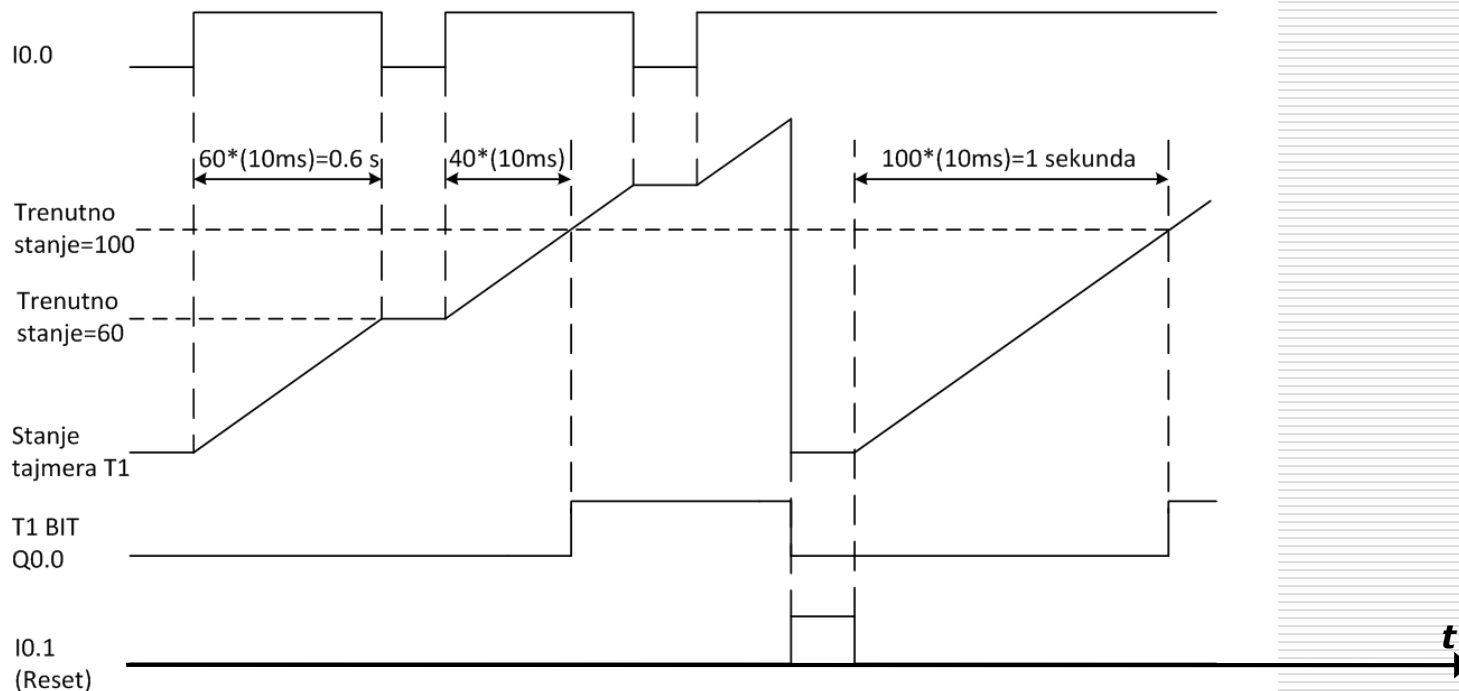
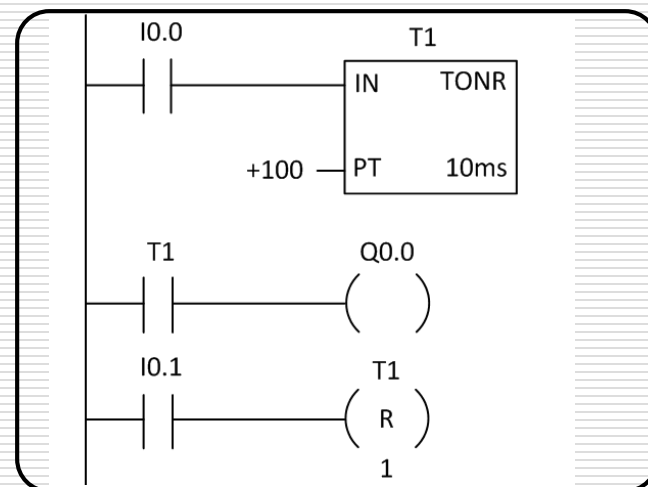


Vremenski dijagram koji opisuje rad TON



Retentive On-Delay Timer (TONR)

TONR je skoro identičan TON. Razlika je što TONR u slučaju ukidanja dozvole brojanja (ulaz IN na nuli) ne resetuje broj nego ga zadržava (pamti). Kada se IN ulaz ponovo aktivira TONR blok nastavlja od vrednosti na kojoj je stao sa brojanjem. Ovim se uvažava akumulacija brojanja vremena tokom višestrukih vremenskih perioda koja je potrebna u nekim primenama. Na primer, ako neka pumpa treba da radi 50 sekundi, ona može da radi prvo 20 sekundi, pa da se njen rad iz nekog razloga suspenduje, pa da se ciklus rada naknadno završi sa dodatnih 30 sekundi rada.



Rezolucija rada TON i TONR kod S-7 serije

Kod S7-200 ima samo određeni broj TON i TONR timera. Ako se želi jedan od ta dva tipa tajmera, mora da se koriste imena iz dole datih tablica

PLC Siemens S7 serija nudi više TON blokova, u različitim rezolucijama, tabela 5-4. Važno je primetiti da brzih *Timer* blokova nema mnogo. Takođe, pošto se brojanje završava sa brojem 32767 duže vremenske intervala je moguće ostvariti sa sporijim *Timer* blokovima.

TON	Rezolucija	Maksimalno vreme koje se može dobiti	Broj <i>Timer</i> bloka
	1 ms	32.767s	T32, T96
	10 ms	327.67s	T33-T36, T97-T100
	100 ms	3276.7s	T37-T63, T101-T255

Tabela 5-4 Različite rezolucije On-Delay tajmera za PLC Siemens S7-200 seriju

Tajmeri TONR tipa su dostupni u tri rezolucije u Siemens S7-200 seriji, tabela 5-5.

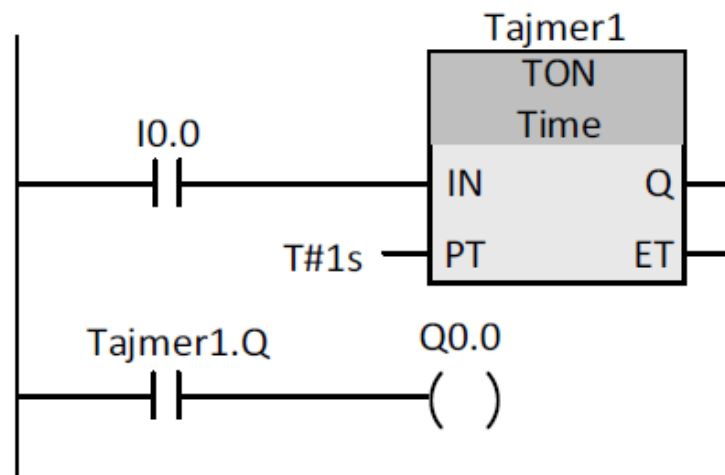
TONR	Rezolucija	Maksimalno vreme	Broj <i>Timer</i> bloka
	1 ms	32.767s	T0, T64
	10 ms	327.67s	T1-T4, T65-T68
	100 ms	3276.7s	T5-T31, T69-T95

Tabela 5-5 Različite rezolucije Retentive On-Delay tajmera za PLC Siemens S7-200 seriju

Realizacija TON bloka u TIA portal (S7-1200)

Moderni S7 PLC imaju malo drugačiji TON blok:

1. Više slobode u izboru imena (**ime**)
2. PT je vreme, a ne broj
3. Izlazni bit je **ime.q**



TON blok u *TIA Portal*-u ima sledeće elemente:

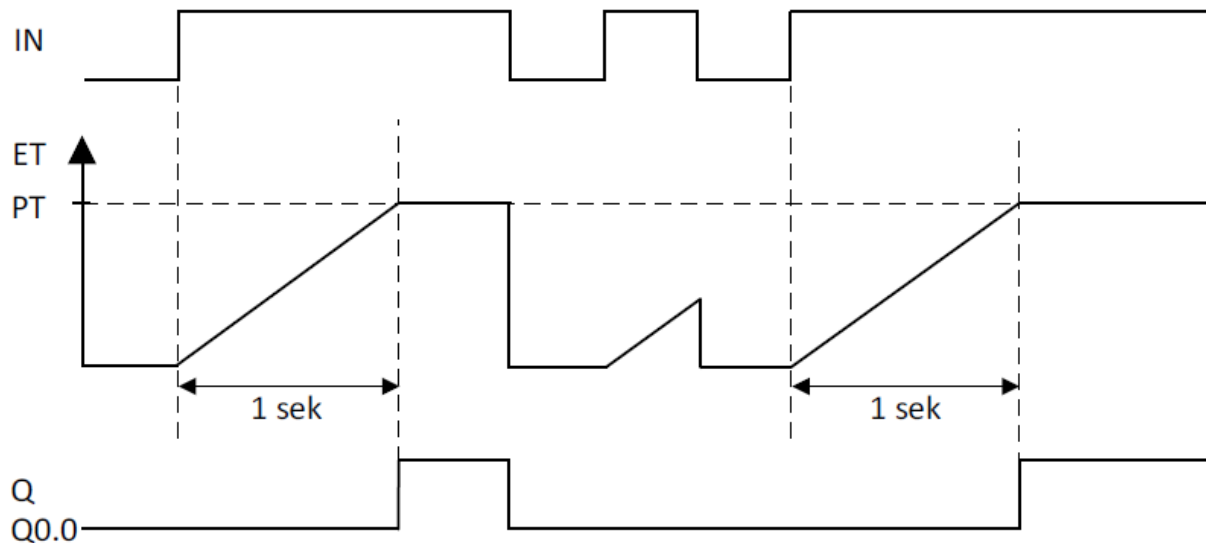
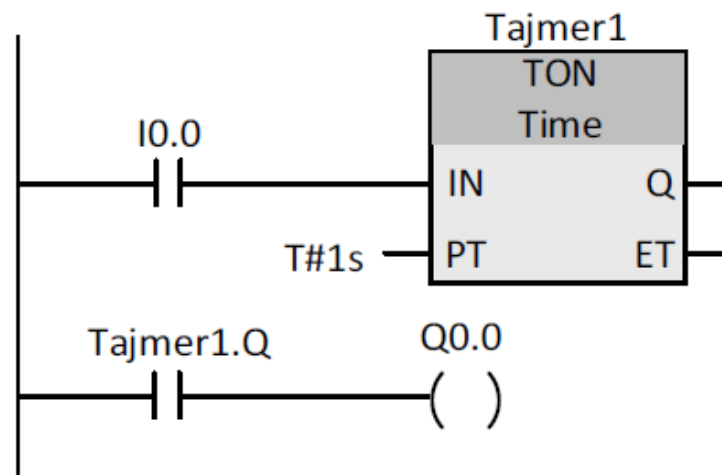
1. Aktivacioni ulaz (IN) koji mu omogućava rad. U datom primeru blok TON meri vreme kada se aktivira uslov ulaza I0.0.
2. Unapred postavljeno vreme koje se meri, PT (*Preset Time*). U datom primeru PT je postavljeno na 1s.
3. Izlaz koji se setuje (Q) kada istekne vreme koje se meri (PT).
4. Trenutno vreme koje se meri može se posmatrati na ET (*Elapsed Time*) izlazu.

Realizacija TON bloka u TIA portal (S7-1200)

Moderni S7 PLC imaju malo drugačiji TON blok:

1. Više slobode u izboru imena (**ime**)
2. PT je vreme, a ne broj
3. Izlazni bit je **ime.q**

Vremenski dijagrami su slični

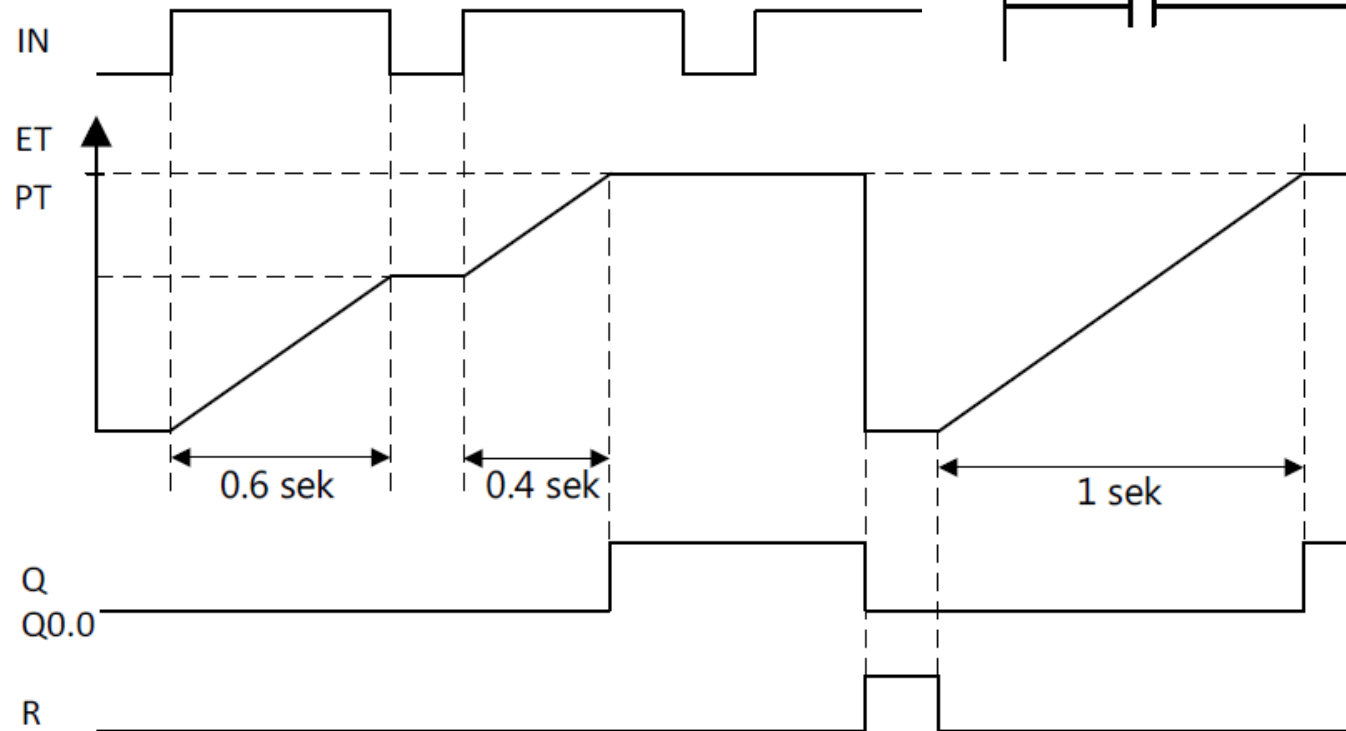
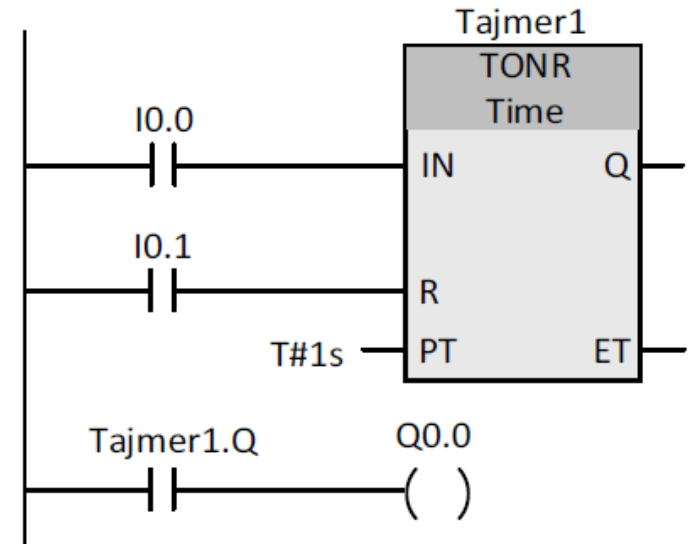


Slika 5.43 Vremenski dijagram koji prikazuje princip rada TON tajmera u TIA Portal-u

Realizacija TONR bloka u TIA portal (S7-1200)

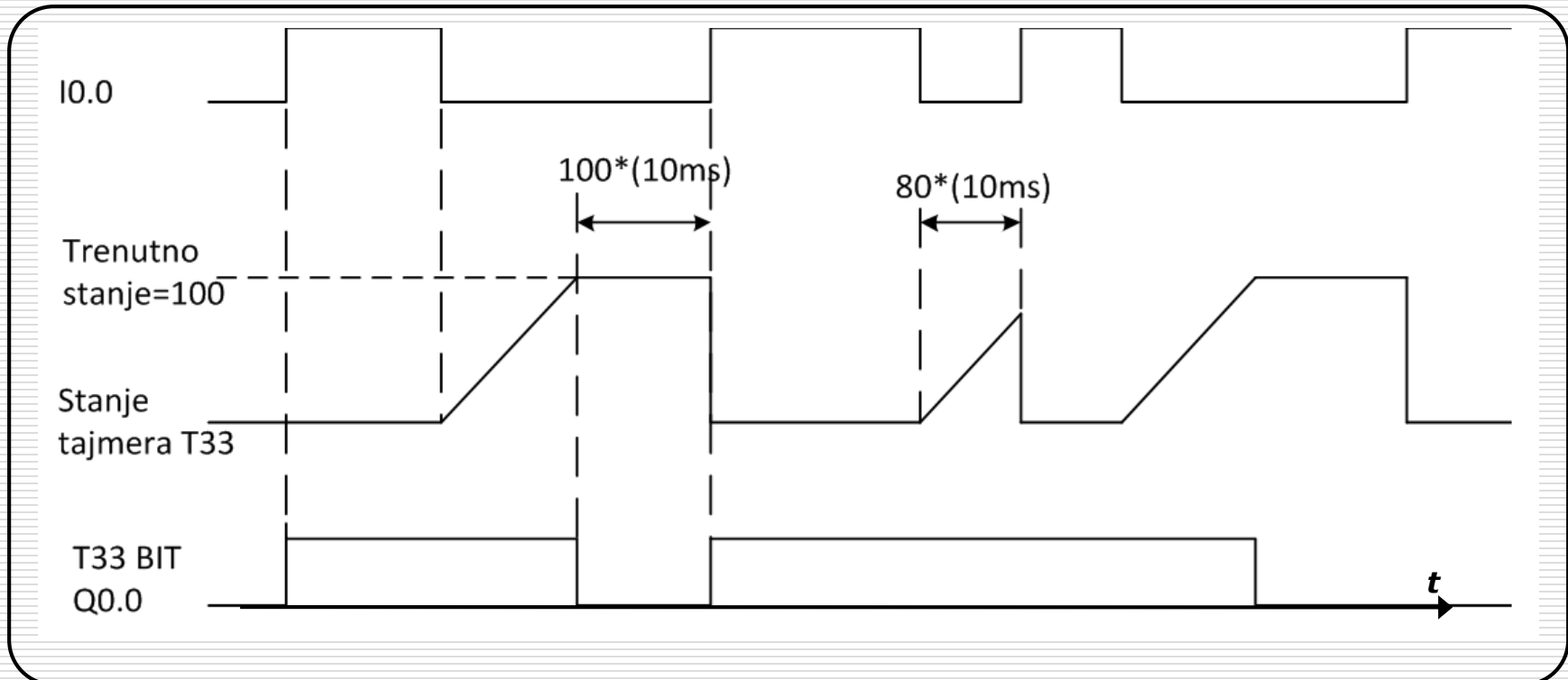
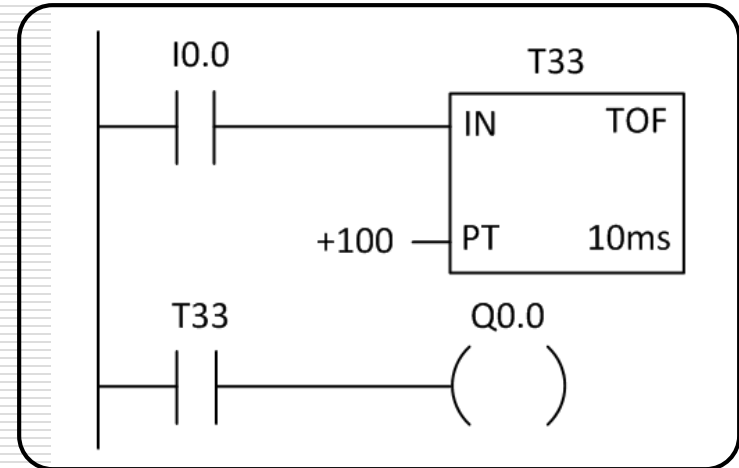
Moderni S7 PLC imaju malo drugačiji TONR blok:

1. Više slobode u izboru imena (**ime**)
2. PT je vreme, a ne broj
3. Izlazni bit je **ime.q**



Tajmer koji kasni isključenje *Off-Delay Timer* (TOF)

Off-Delay tajmerski blok se koristi da se nakon isključenja nekog ulaza za neko vreme odloži isključenje nekog izlaza. Uključenje se ne odlaže. Čim se aktivira ulaz IN tajmerski bit T33 na slici 5.30 se postavlja na 1. Ali, kada se IN deaktivira ($I0.0=0$), TOF broji vremenske intervale i tek kada se postigne PT vreme spušta vrednost T33 na 0. Ako se ulaz IN pre isteka vremena ponovo aktivira, T33 ostaje jednak jedinici, a brojanje se suspenduje.



Tajmer koji kasni isključenje *Off-Delay Timer* (TOF)

Kod S7-200 ima samo određeni broj TOF timera. Ako se želi taj tip tajmera, mora da se koriste imena iz dole date tablice

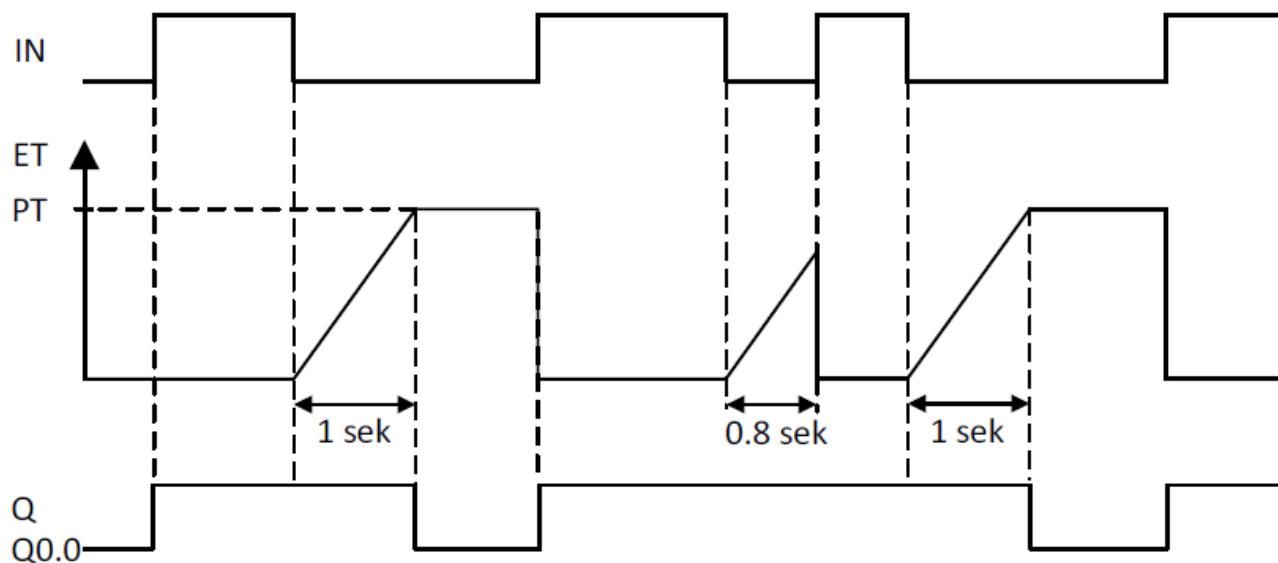
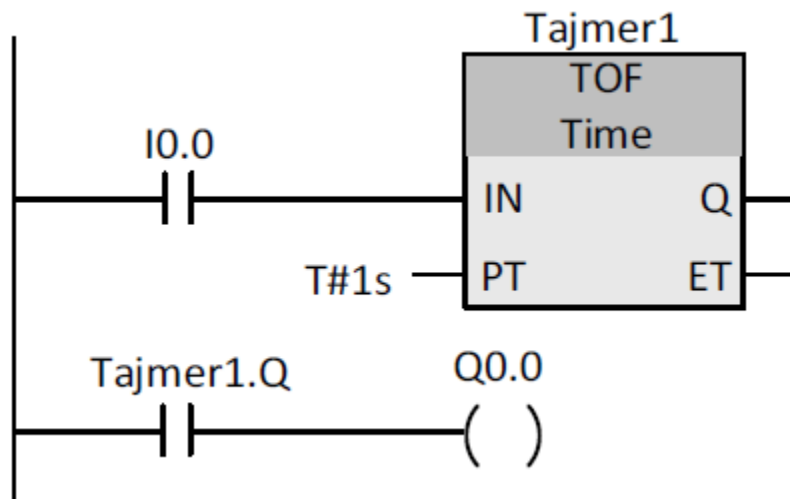
TOF	Rezolucija	Maksimalno vreme	Broj <i>Timer</i> bloka
	1 ms	32.767s	T32, T96
	10 ms	327.67s	T33-T36, T97-T100
	100 ms	3276.7s	T37-T63, T101-T255

Tabela 5-6 Različite rezolucije Off-Delay tajmera u slučaju Siemens S7-200 serije PLC-a

Realizacija TOF bloka u TIA portal (S7-1200)

Moderni S7 PLC imaju malo drugačiji TOF blok:

1. Više slobode u izboru imena (**ime**)
2. PT je vreme, a ne broj
3. Izlazni bit je **ime.q**



Slika 5.51 Vremenski dijagram koji prikazuje princip rada TOF tajmera u TIA Portal-u

Blokovi za merenje broja impulsa (Counter blokovi)

Čemu služe Counter blokovi u PLC programu?

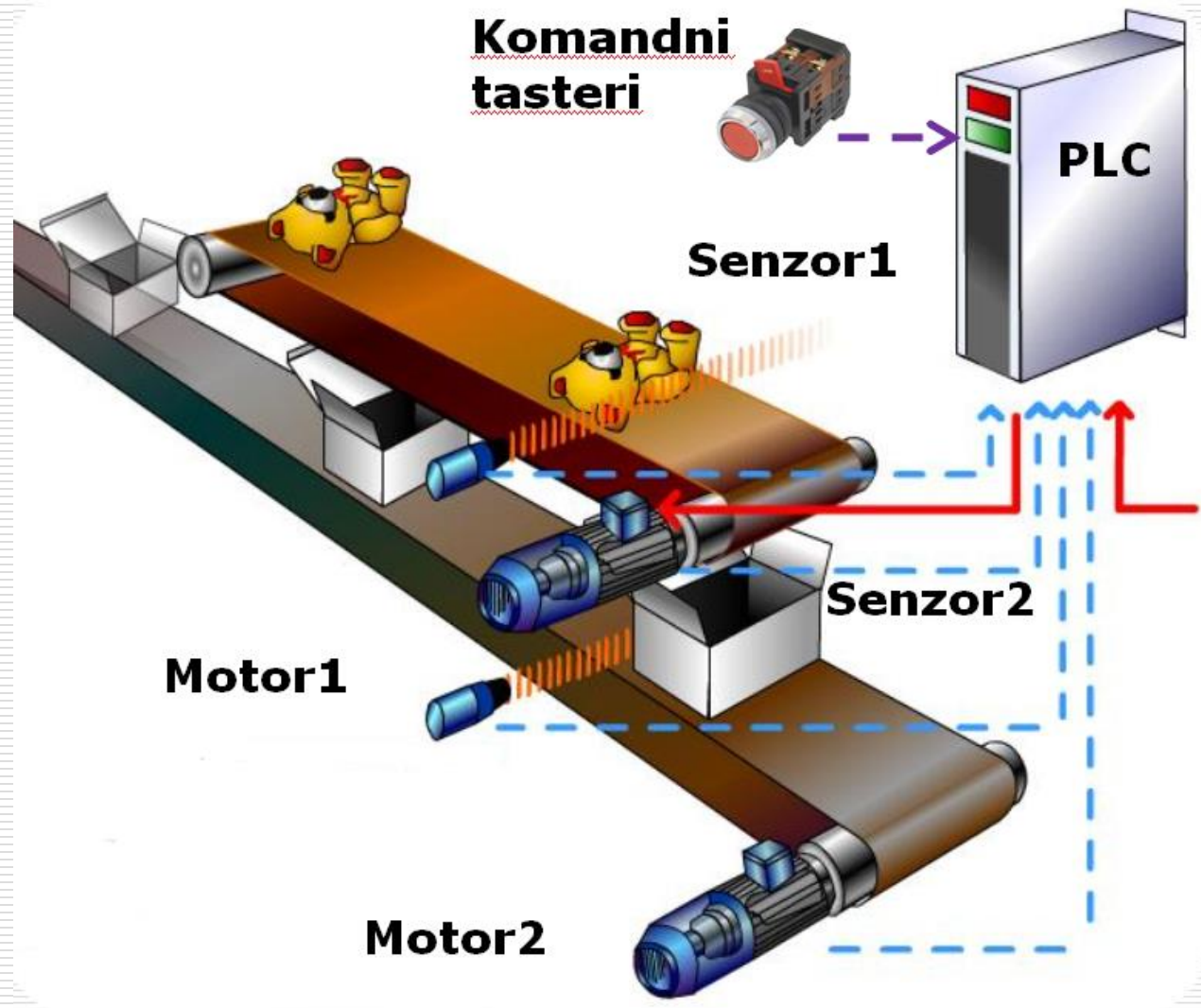
Kontrola pakovanja - pakerica

Senzor1 javlja koliko je prošlo proizvoda.

Senzor2 javlja kada je nova kutija dosla na mesto.

PLC na osnovu tih signala i komandi (**tasteri**) upravlja radom **motor1** i **motor2**

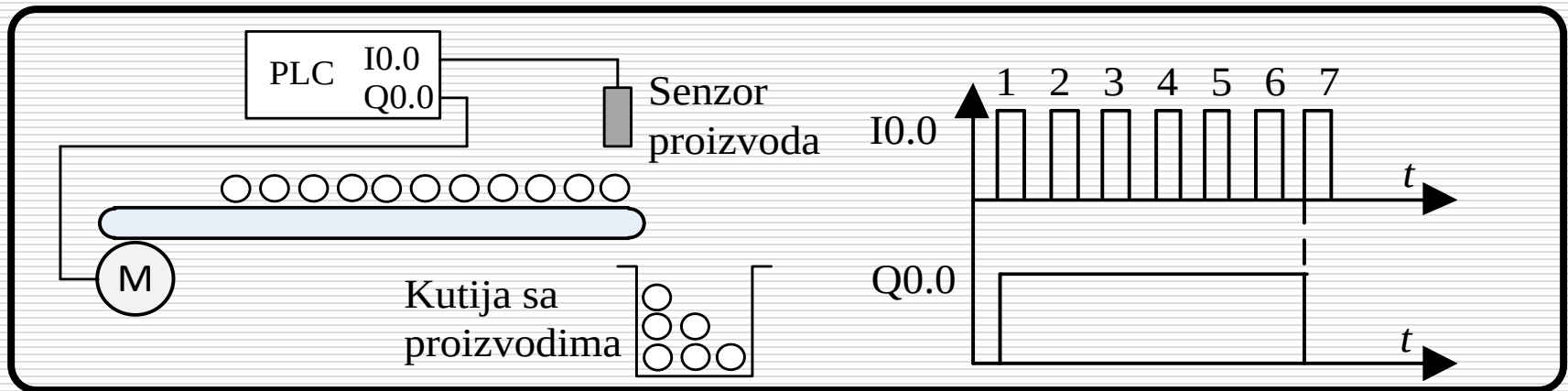
KO ĆE DA
BROJI
MEDVEDIĆE



Čemu služe Counter blokovi u PLC programu?

Brojanje impulsa ili događaja je jedan od važnijih zadataka tokom upravljanja industrijskim sistemima.

Primer, PLC upravlja motorom pokretne trake tako što uključi motor, zatim broji proizvode koji ulaze u kutiju i kada izbroji 7 proizvoda koja su ušla zaustavlja motor, tj. traku.



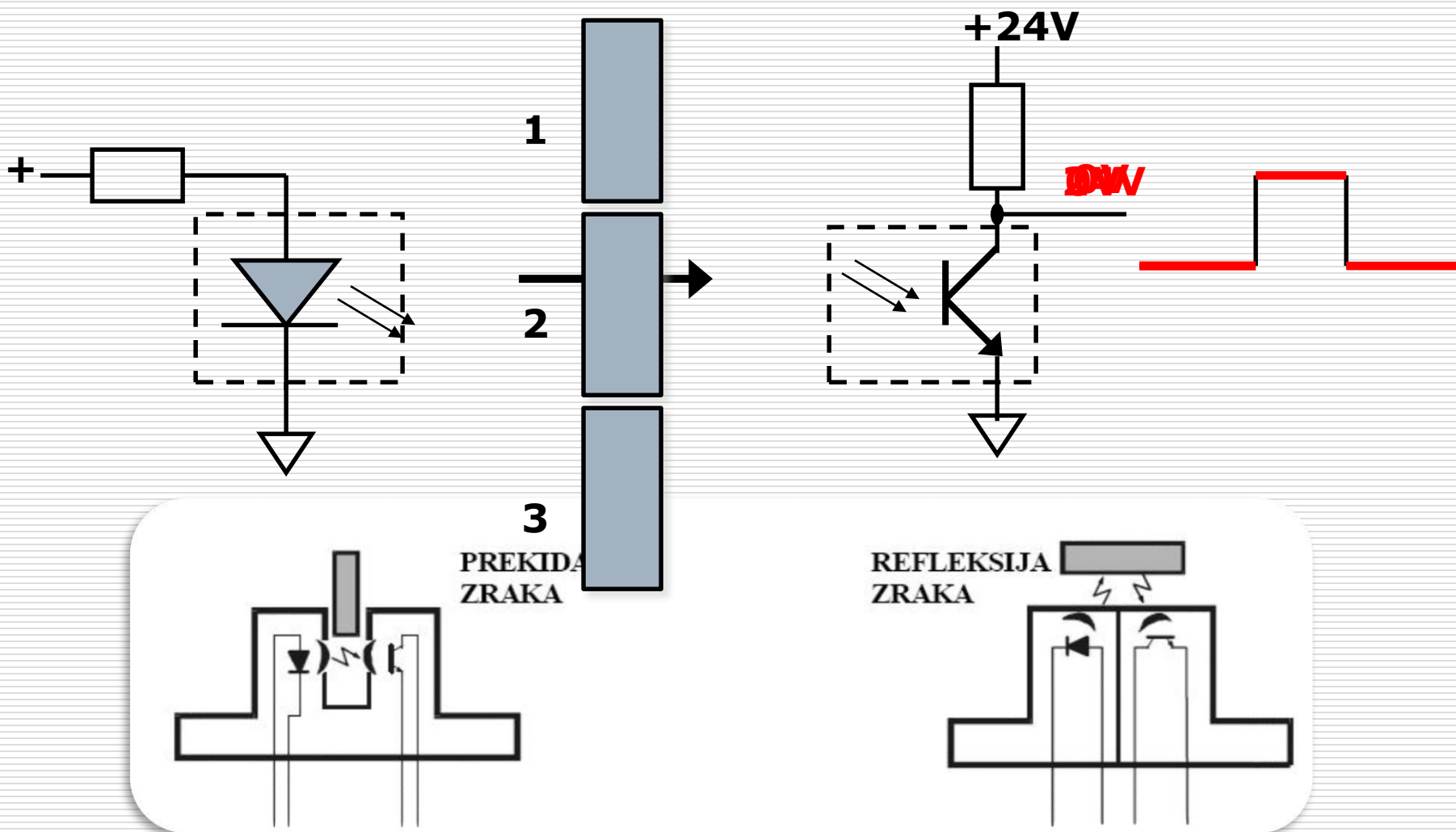
Da bi PLC ovaj zadatak precizno izvršio neophodno je da poseduje blok za brojanje impulsa. U tu svrhu PLC poseduju brojačke (*Counter*) blokove koji broje impulse detektovane na svom ulazu. Ulaz *Counter* bloka se može povezati sa PLC ulazom (na slici 5.32 sa ulazom I0.0) ali i povezati za bilo koji drugi bit ili logiku u programu.

Primer prostih impulsnih senzora

Princip rada impulsnih svetlosnih senzora

Predajnik, svetlosni izvor, šalje svetlost u pravcu prijemnika.

Prijemnik menja stanje izlaza svaki put kada se svetlosni snop prekine.



Koje Counter blokove PLC poseduje ?

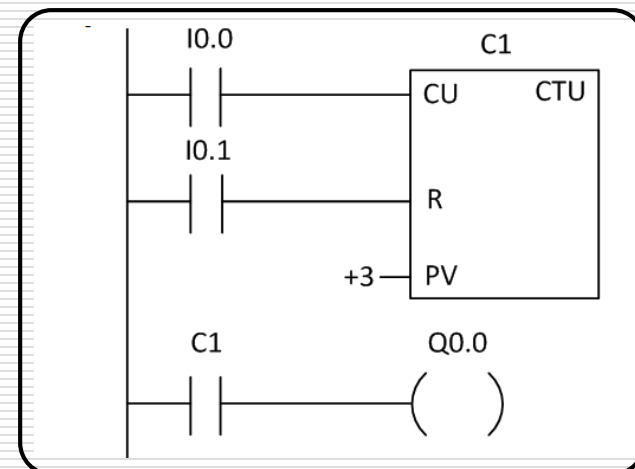
PLC poseduje tri osnovne grupe *Counter* blokova:

- 1. brojače nagore (*Counter Up*),**
- 2. brojače nadole (*Counter Down*) i**
- 3. brojače nagore/nadole (*Count Up/Down*).**

Brojač je zamišljen da broji impulse nadošle na nekom PLC ulazu. Ali, u opštem slučaju brojački blok može da broji promene bilo kog bita unutar PLC programa.

Brojač nagore (*Count Up - CTU*)

Brojač je nagore i zato uvećava svoju vrednost svaki put kada CU ulaz pravi tranziciju od 0 na 1 (u datom primeru reaguje na rastuću naponsku ivicu ulaznog signala preko uslova I0.0). Kada je trenutna vrednost brojača jednaka ili veća od postavljene vrednosti PV (u ovom primeru na PV ulaz je postavljen broj 3), brojač bit (ovde C1) će se uključiti (C1=1).

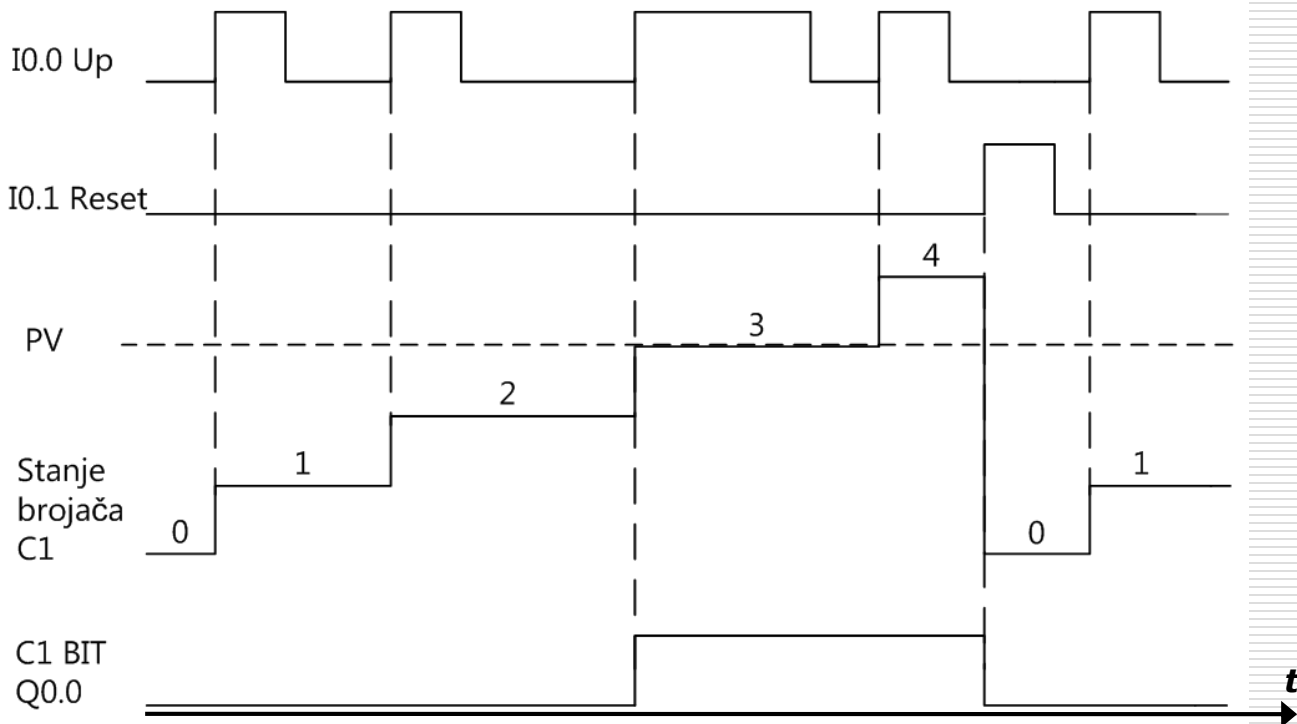
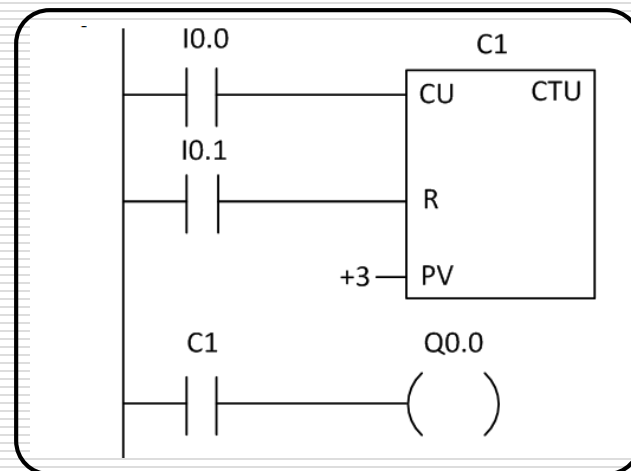


CTU blok broji ima sledeće elemente:

1. Brojački ulaz (CU) na koji se dovode impulsi koje treba brojati. Svaki put kada CU ulaz pravi tranziciju iz stanja 0 u stanje 1 vrednost brojača se uveća za jedan (CTU). U primeru na slici uvećanje se dešava kada se napon na PLC ulazu I0.0 promeni sa 0V na 24V.
2. Unapred postavljeni broj impulsa koji se meri, PV (*Preset Value*). U datom primeru na ulaz PV je postavljen broj 3.
3. Bit koji taj *Counter* blok kontroliše, u datom primeru je uzet brojač C1 koji kontroliše bit C1. Ovaj bit se može koristiti u ostatku programa kao signal da je izbrojan potreban broj impulsa. U datom primeru je iskorišćen u instrukcijskoj liniji ispod brojačkog bloka.
4. Brojač se resetuje kada se uključi R ulaz (ulaz *Reset* instrukcije), ili kada se izvrši *Reset* instrukcija nad bitom brojača, u ovom slučaju C1.

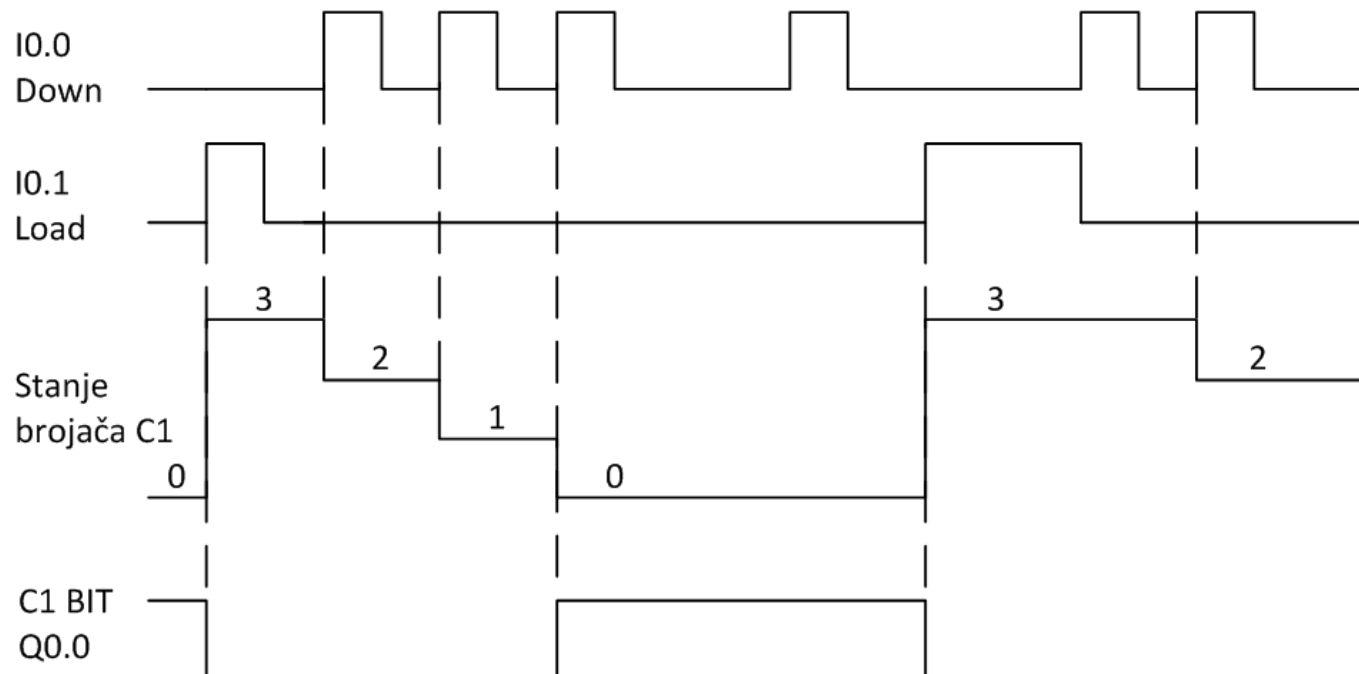
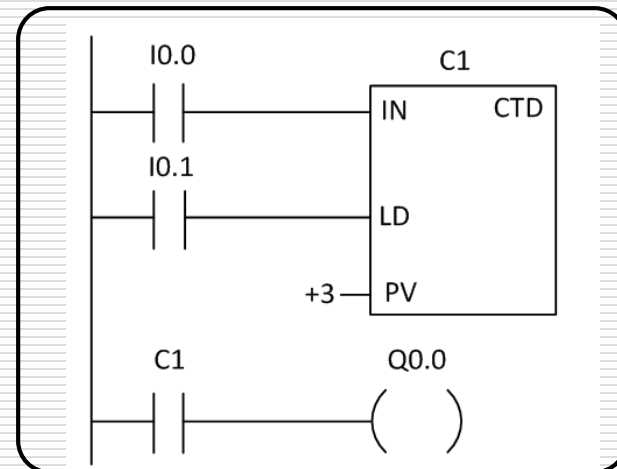
Brojač nagore (*Count Up - CTU*)

Brojač je nagore i zato uvećava svoju vrednost svaki put kada CU ulaz pravi tranziciju od 0 na 1 (u datom primeru reaguje na rastuću naponsku ivicu ulaznog signala preko uslova I0.0). Kada je trenutna vrednost brojača jednaka ili veća od postavljene vrednosti PV (u ovom primeru na PV ulaz je postavljen broj 3), brojač bit (ovde C1) će se uključiti (C1=1).



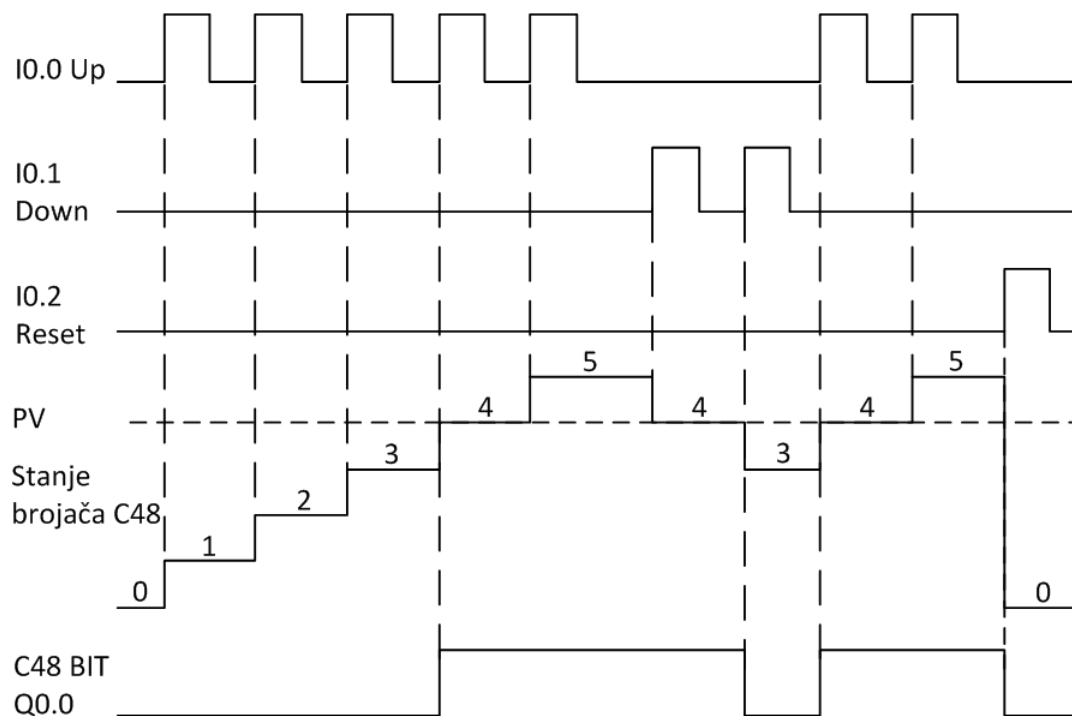
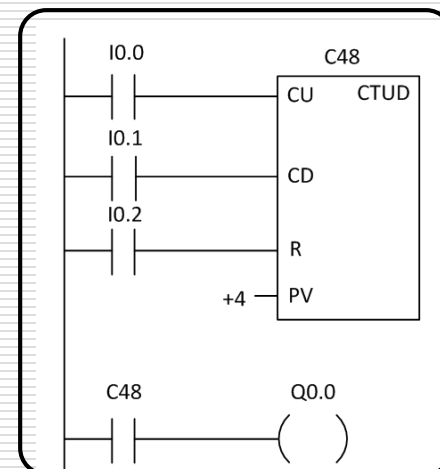
Brojač nadole (*Count Down - CTD*)

CTD broji nadole od svoje trenutne vrednosti, svaki put kada se na ulazu CD detektuje tranzicija sa 0 na 1. Kada vrednost brojača dostigne 0, brojač bit Cx se uključuje (Cx=1) i zaustavlja se brojanje. U datom primeru C1=1 omogućuje izvršavanje naredbe Q0.0, tj. izlaz se setuje. Kada se LD ulaz uključi (LD=1) brojač resetuje brojač bit i učitava postavljenu vrednost PV (u ovom primeru +3). Resetovanjem brojač bita C1 prekida se izvršenje instrukcije Q0.0.



Brojač nagore/nadole (*count up/down , CTUD*)

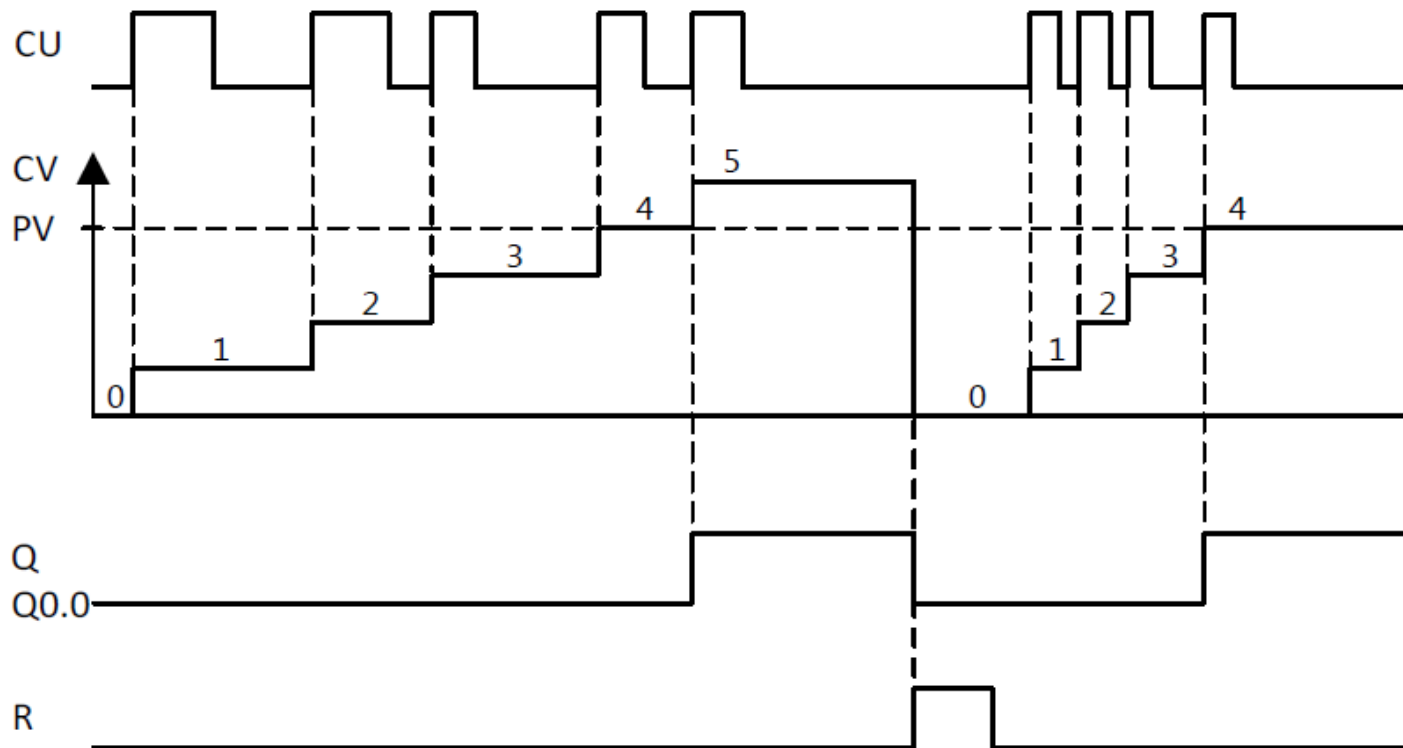
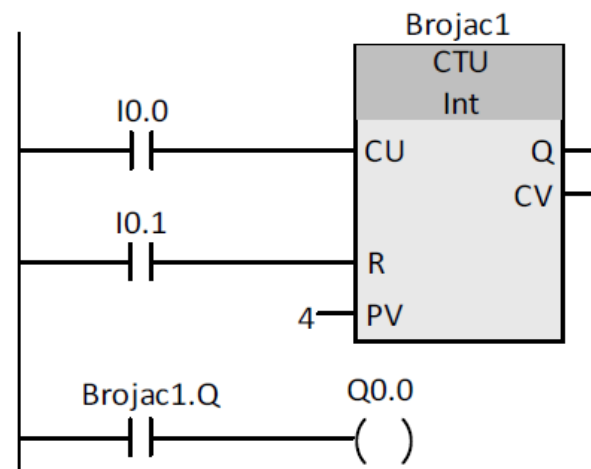
CTUD svaki put kada CU ulaz pravi tranziciju sa 0 na 1 uvećava, a kada CD ulaz pravi tranziciju umanjuje svoju vrednost. U primeru brojač uveća vrednost na rastuću ivicu ulaznog signala preko uslova I0.0 a umanjuje je na rastuću ivicu ulaznog signala na I0.1. Trenutna vrednost brojač bita (u datom primeru C48) odražava trenutno stanje brojača u odnosu na postavljenu vrednost PV. Ako je vrednost brojača jednaka ili veća od upisane u PV bit C48 je setovan, ako je manja bit C48 je resetovan.



Realizacija CTU bloka u TIA portal (S7-1200)

Moderni S7 PLC imaju malo drugačiji CTU blok:

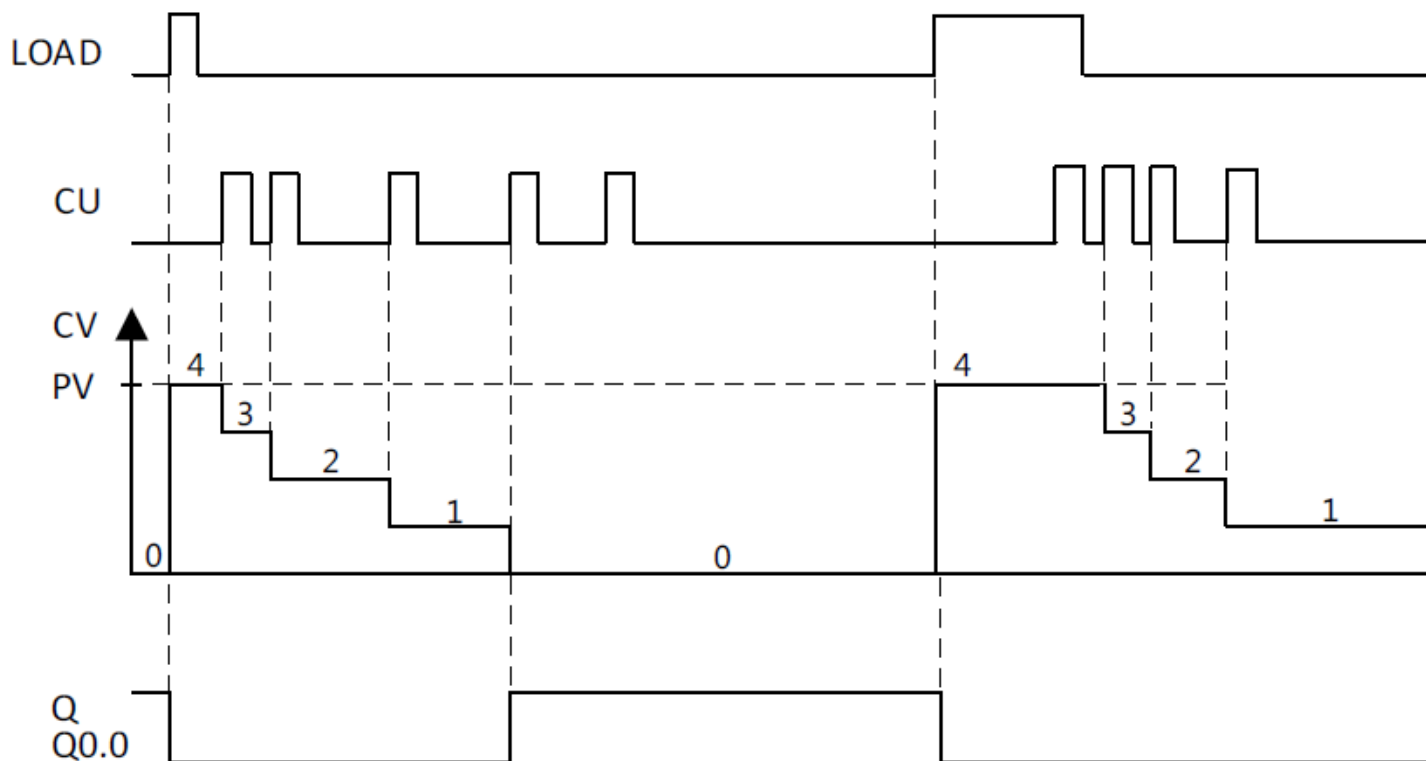
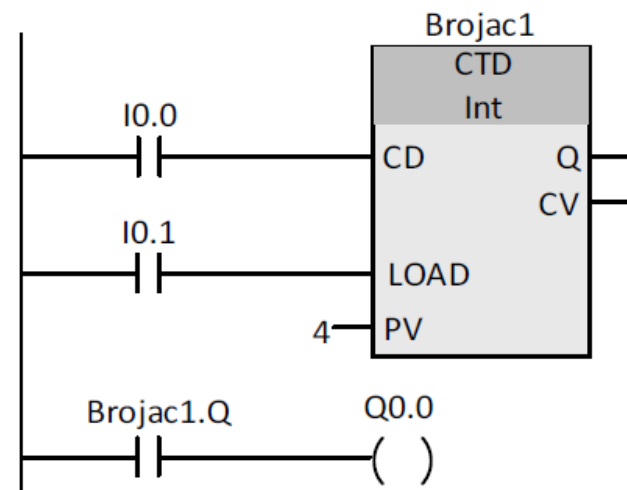
1. Više slobode u izboru imena (**ime**)
2. Izlazni bit je **ime.q**



Realizacija CTD bloka u TIA portal (S7-1200)

Moderni S7 PLC imaju malo drugačiji CTD blok:

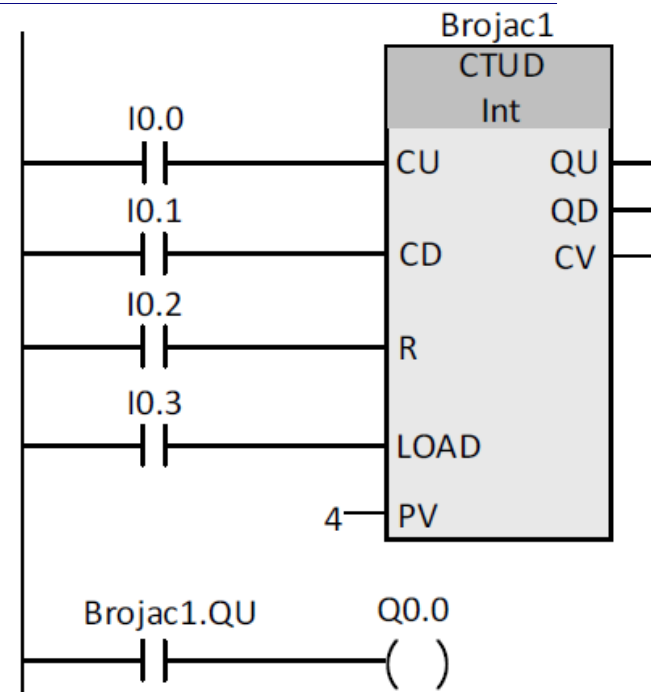
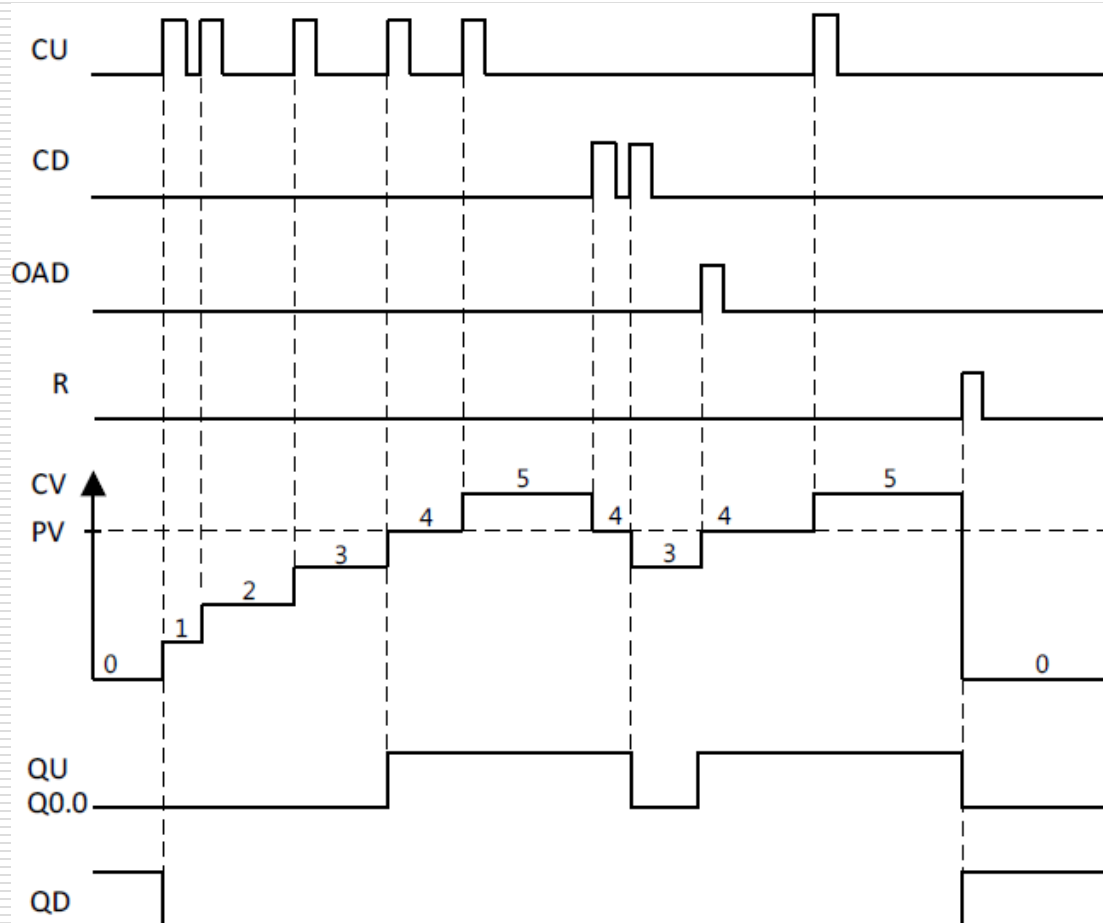
1. Više slobode u izboru imena (**ime**)
2. Izlazni bit je **ime.q**



Realizacija CTUD bloka u TIA portal (S7-1200)

Moderni S7 PLC imaju malo drugačiji CTUD blok:

1. Više slobode u izboru imena (**ime**)
2. Izlazni bit je **ime.q**



Specijalni blokovi za upravljanje energetskim pretvaračima

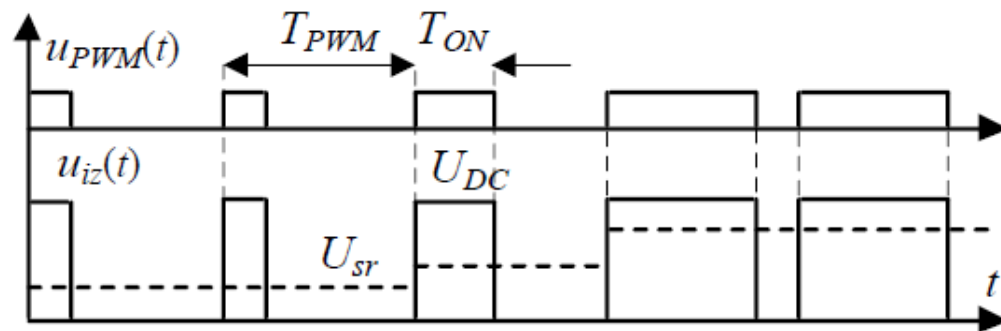
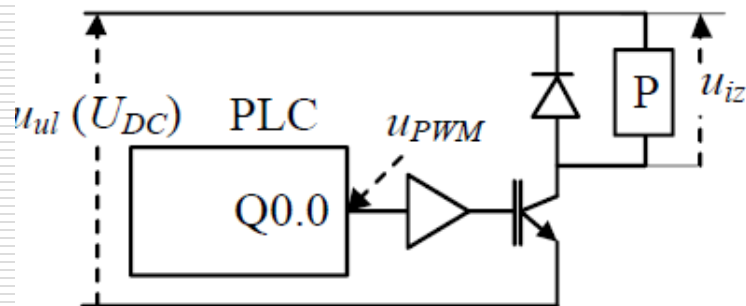
Blok za generisanje PWM signala

Impulsno širinska modulacija (IŠM) = Pulse Width Modulation (PWM)

PWM signal ima srednju vrednost jednaku željenoj vrednosti.

$$U_{sr} = \frac{1}{T_{PWM}} \int_0^{T_{PWM}} u_{PWM}(t) dt = \frac{T_{ON}}{T_{PWM}} U_{DC}$$

Kako PLC može da ga generiše?



Perioda PWM signala može biti mnogo kraća od jednog skan ciklusa

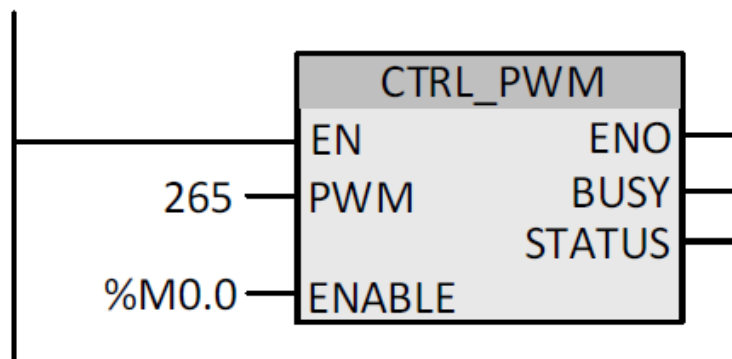
Blok za generisanje PWM signala

Perioda PWM signala može biti mnogo kraća od jednog skan ciklusa
Zato se koristi PWM blok koji nezavisno od programa generiše PWM signal, zadate periode i vremena uključenja, na nekom od PLC izlaza.

PWM izlaz treba konfigurisati u TIA portalu

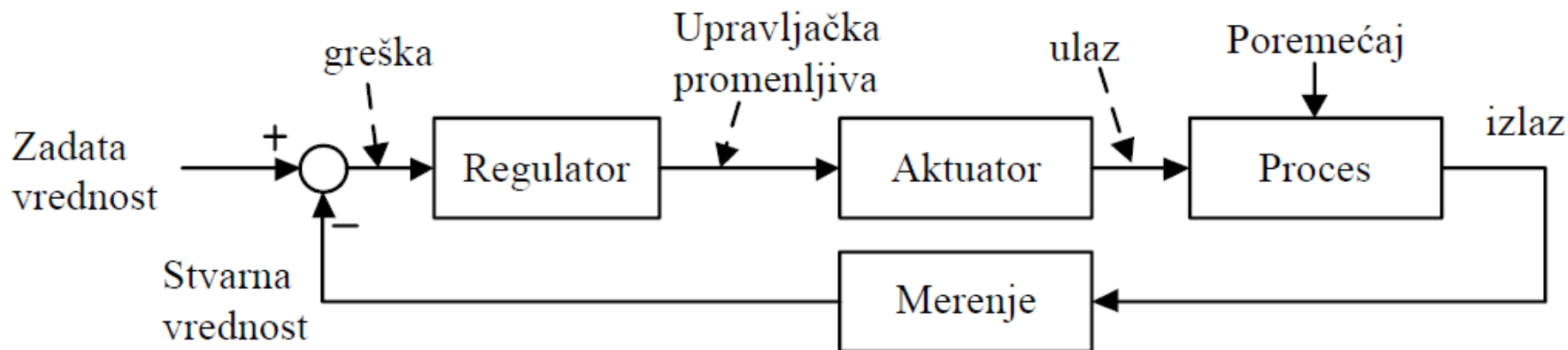
The screenshot shows the 'Pulse options' and 'Hardware identifier' configuration windows in the TIA Portal. In the 'Pulse options' window, the 'Signal type' is set to 'PWM', 'Time base' to 'Milliseconds', 'Pulse duration format' to 'Hundredths', 'Cycle time' to '100 ms', and 'Initial pulse duration' to '50 Hundredths'. In the 'Hardware identifier' window, the 'Hardware identifier' is set to '265'. Below these, the 'Hardware outputs' section shows the 'Pulse output' set to '%Q0.0'.

Zatim je potrebno PWM blok uključiti u program i dozvoliti mu rad



Blok PID regulator

Ukoliko je potrebno precizno regulisati neku brzinu neophodna je povratna sprega i regulator

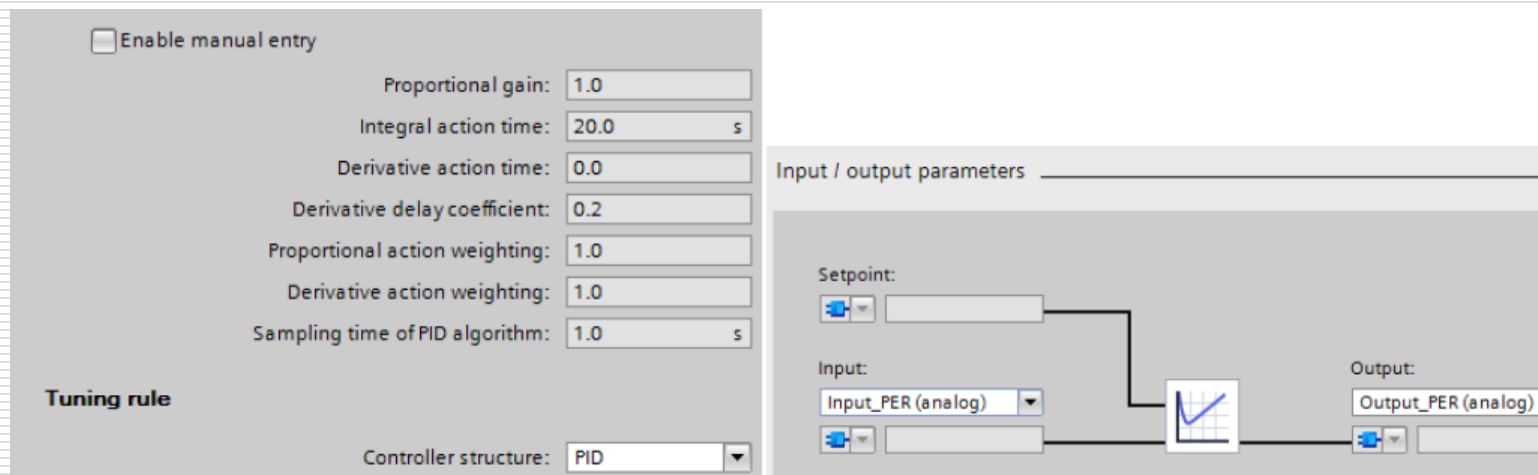


PID – svaka akcija ima svoje prednosti

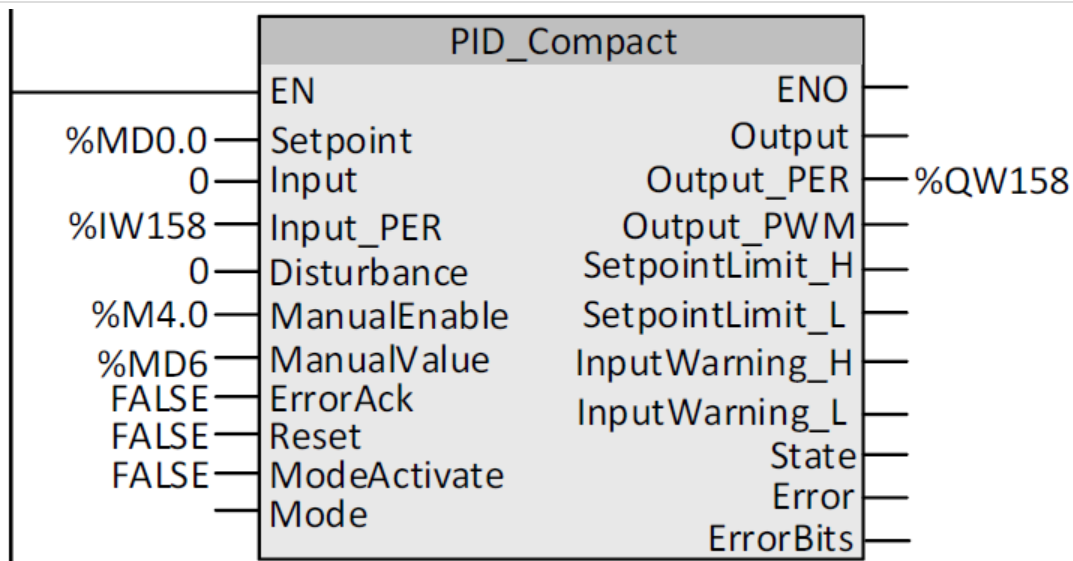
$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de}{dt}$$

Blok PID regulator

PID regulator treba konfigurirati u TIA portalu



Zatim ga je potrebno uključiti u program i dozvoliti mu rad



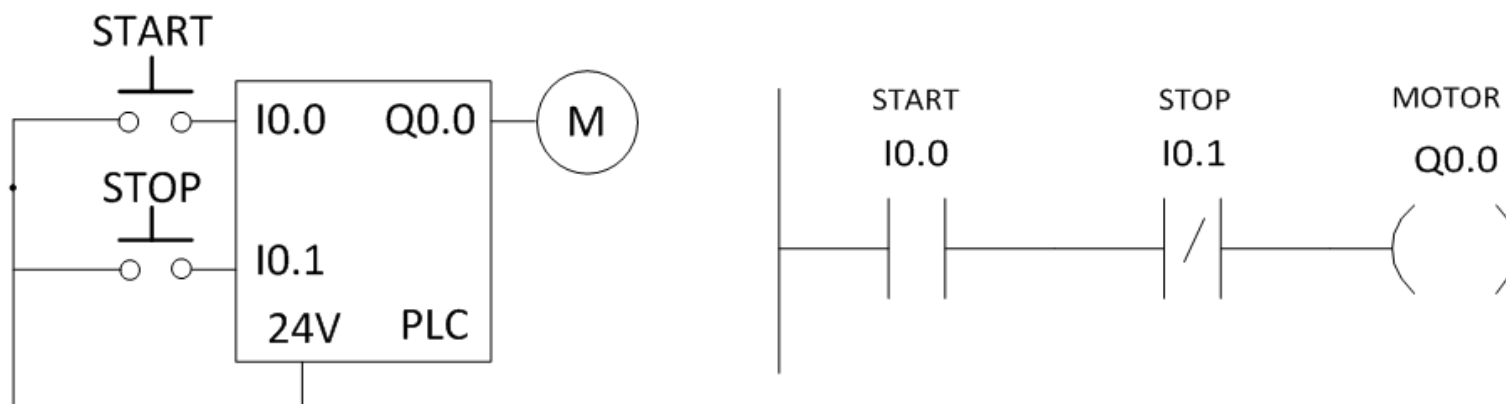
1. zadana vrednost izlaza (SetPoint) se uzima iz memorijske lokacije MD0.0,
2. merena vrednost izlaza (Input_PER) se uzima sa analognog ulaza IW158.
3. upravljačka prom. (Output_PER) se u ovom primeru šalje na analogni PLC izlaz QW158 koji je pretvara u napon koji se dalje šalje kao referentna vrednost ka nekom aktuatoru. Ovde je moguće i direktno slati PWM signal iz PLC, u tom slučaju se koristi opcija Output_PWM,

Primeri Ladder dijagrama

Primer samodržanja kontakata

Kao primer je uzeto da PLC upravlja motorom (M) koji je kontrolisan sa izlazom Q0.0. START taster je izveden sa NO kontaktima i povezan je na ulaz kontrolera I0.0. Na drugi kontakt START tastera je povezano 24V napajanje za senzore (L+) tako da se njegovim pritiskom na I0.0 pojavljuje napon od 24V tj. logička jedinica.

U prvom (lošem) rešenju uzimamo i STOP taster sa NO kontaktima i da je povezan na ulaz I0.1. Na drugi kontakt STOP tastera je takođe povezano 24V napajanje za senzore (L+).



Slika 5.13 Šema ožičenja PLC ulaza i izlaza, loš primer upravljanja motorom

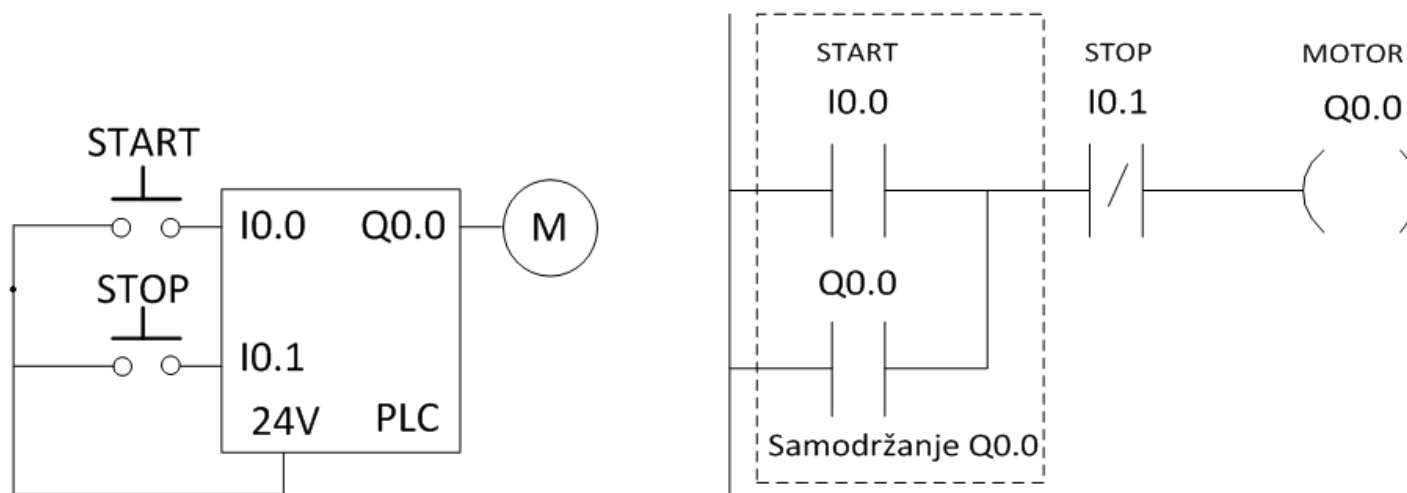
Po pritisku START tastera motor treba da počne da radi. U isto vreme STOP taster ne treba da bude pritisnut. Da bi se izlaz Q0.0 aktivirao pri ovim uslovima u PLC programu je zadata instrukcija prikazana na slici 5.13 desno. Izlaz Q0.0 je 1 ako je $I0.0=1$ i $I0.1=0$, tj. oba uslova ispunjena.

Problem sa ovim rešenjem je što otpuštanje START tastera dovodi do situacije $I0.0=0$ i $I0.1=0$ u kojoj prvi uslov nije ispunjen i blokira Q0.0 koji postaje nula. Radnik koji manipuliše radom ove mašine bi nakon nekoliko sati držanja START tastera prilično zamrzio PLC programera.

Primer samodržanja kontakata

U drugom (boljem, ali i dalje lošem) rešenju realizujemo samodržanje izlaza.

Samodržanje kontakta se u PLC programu ostvaruje tako što se paralelno sa uslovom za uključenje izlaza postavlja radni NO uslov kontrolisan od istog tog izlaza. **Princip samodržanja:** Nakon što je pobuđen, IZLAZ „drži samog sebe“ u pobuđenom stanju (START signal više nije neophodan). U konkretnom slučaju samodržanje se realizuje tako što se NO uslov START tastera i NO uslov izlaza Q0.0 postavljaju paralelno, slika 5.14. Prvim pritiskom na taster START izlaz Q0.2 se aktivira i bit Q0.2 postaje 1. Sada je moguće otpustiti START taster jer je uslov $Q0.2=1$ zadovoljen i time izlaz Q0.2 „samog sebe održava“.

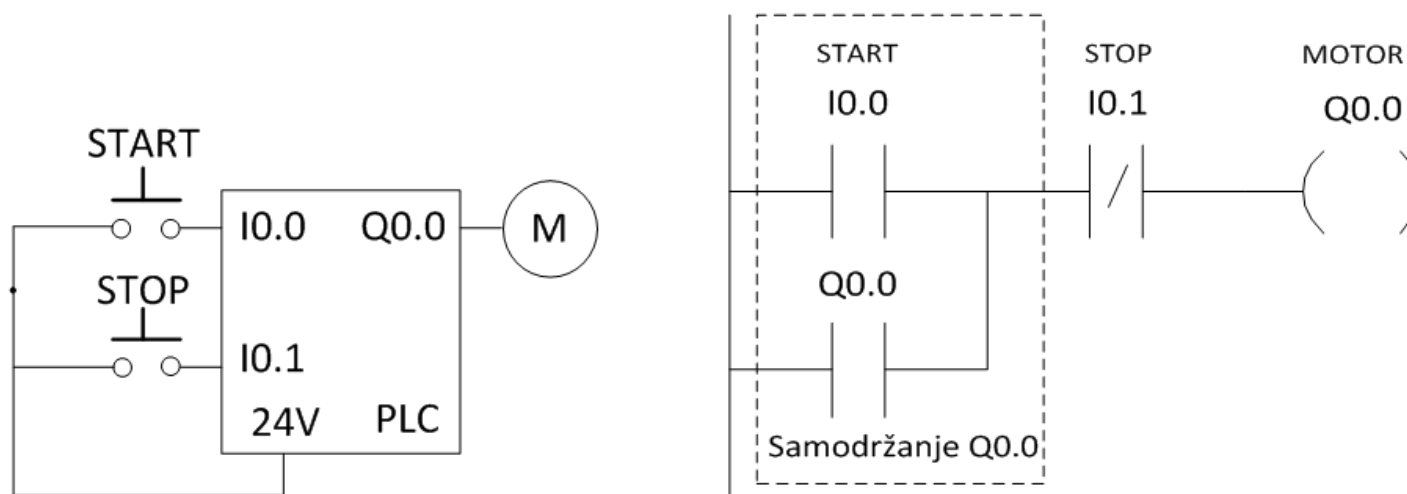


Slika 5.14 Šema ožičenja PLC ulaza i izlaza, i dalje loš primer upravljanja motorom

Primer samodržanja kontakata

U drugom (boljem, ali i dalje lošem) rešenju realizujemo samodržanje izlaza.

Motor treba da se zaustavi pritiskom na STOP taster koji kontroliše uslov na red u liniji instrukcije i po potrebi blokira izlaz. Motor se zaustavlja kada se taster STOP pritisne, što zatvara njegove NO kontakte i dovodi napona na ulaz I0.1. Time uslov $I0.1=0$ postaje neaktivan, blokira prolaz i postavlja i konačni uslov linije instrukcija Q0.0 na nivo 0, slika 5.14. Motor je ovim zaustavljen a ujedno je i deaktivirano samodržanje, tj. instrukcija Q0.0 koja je paralelna sa START instrukcijom. Sada se motor neće ponovo uključiti po deaktiviranju tastera STOP. Motor se može ponovo pokrenuti samo novim pritiskom na START taster.

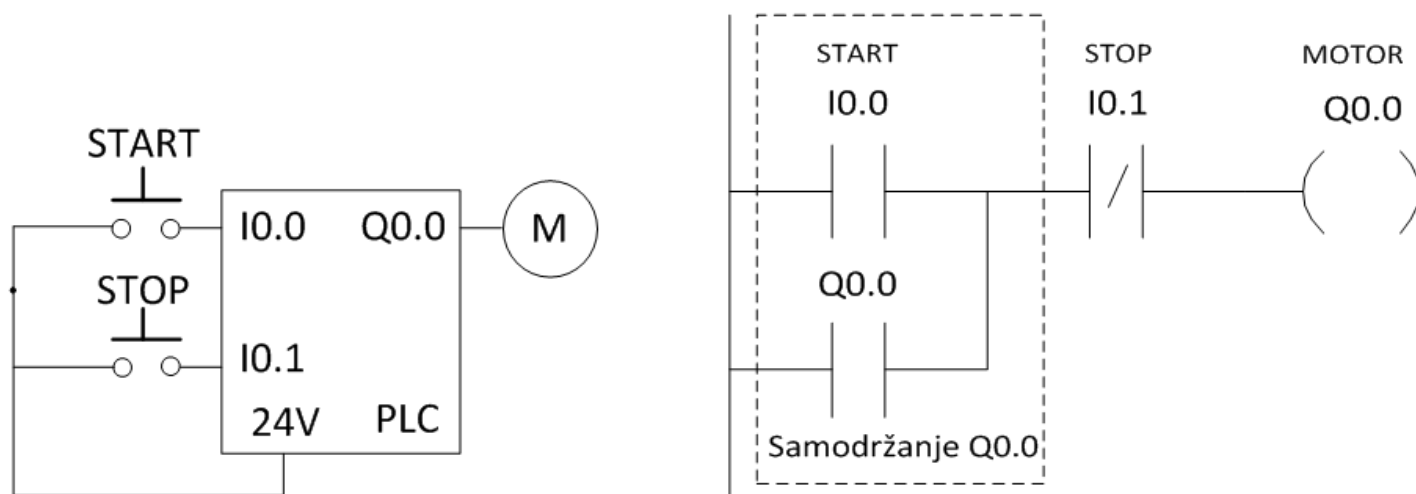


Slika 5.14 Šema ožičenja PLC ulaza i izlaza, i dalje loš primer upravljanja motorom

Primer samodržanja kontakata

U drugom (boljem, ali i dalje lošem) rešenju realizujemo samodržanje izlaza.

Problem kod ove šeme je sigurnosne prirode. Motor se zastavlja ako se pritisne taster STOP, tj. dovede 24V na ulaz I0.1. A šta ako nema tog napona, jer se na primer prekinula žica? Tada operater uzaludno pritiska taster STOP jer se napon na I0.1 ne menja. Ovo je opasna situacija sa kvarom u kontrolnom kolu, a motor ne može da bude zaustavljen.

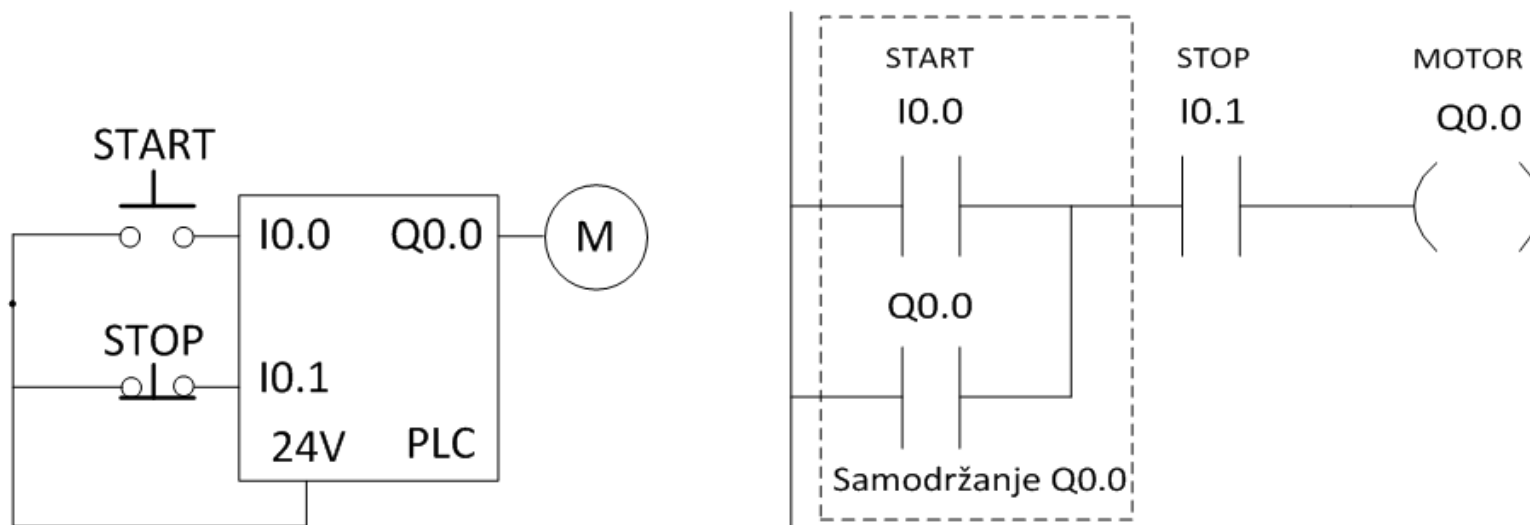


Slika 5.14 Šema ožičenja PLC ulaza i izlaza, i dalje loš primer upravljanja motorom

Primer samodržanja kontakata

Konačno rešenje

Iz sigurnosnih razloga STOP taster se realizuje sa tasterom sa normalno zatvorenim (NC) kontaktima. Prednost korišćenja STOP tastera sa NC kontaktima je u tome da se motor zaustavlja bilo da je NC taster pritisnut, tj. kontakti otvoreni, bilo da je došlo do nestanka kontrolnog napona. U prvom slučaju motor se zaustavlja usled dobijene STOP komande, u drugom slučaju usled kvara i nemogućnosti rada kontrolnog kola.



Slika 5.15 Šema ožičenja PLC ulaza i izlaza, dobar primer upravljanja motorom